

Accountable Systems:
Enabling Appropriate Use of Information on the
Web

by

Oshani Wasana Seneviratne

S.M., Massachusetts Institute of Technology (2009)

B.Sc. (Hons), University of Moratuwa, Sri Lanka (2007)

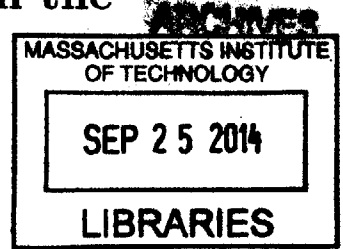
Submitted to the Department of Electrical Engineering and Computer
Science

in partial fulfillment of the requirements for the degree of
Doctor of Philosophy in Computer Science and Engineering
at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2014

© Massachusetts Institute of Technology 2014. All rights reserved.



Signature redacted

Author

Department of Electrical Engineering and Computer Science

August 29, 2014

Signature redacted

Certified by.....

Tim Berners-Lee

Professor

Thesis Supervisor

Signature redacted

Certified by.....

Lalana Kagal

Principal Research Scientist

Thesis Supervisor

Signature redacted

Accepted by

Leslie A. Kolodziejcki

Chairman, Department Committee on Graduate Theses



**Accountable Systems:
Enabling Appropriate Use of Information on the Web**

by

Oshani Wasana Seneviratne

Submitted to the Department of Electrical Engineering and Computer Science
on August 29, 2014, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Computer Science and Engineering

Abstract

The Web is plagued by problems of privacy and piracy. In each instance, outdated laws combined with current technology provides little reassurance to information providers, and may have damaging side effects. To meet this challenge, we have designed, built, and tested and present a new architecture for information exchange on the Internet called HTTPPA (Hyper Text Transfer Protocol with Accountability). In this 'Accountable' architecture, information use is tracked from its creation through its modification, repurposing and republishing with the help of the 'Provenance Tracking Network', a decentralized network of peers that together record the rules governing resources on the Web, coupled with how these resources are shared and used.

We found that the accountable systems framework provides an attractive compromise where the rights and abilities of parties to control access and use is balanced against the burden of restrictions imposed for two prototype applications; one dealing with privacy in healthcare, and the other with rights in photo sharing. Healthcare patients given the ability to be notified of use of their medical records judged that they had sufficient privacy protection, while doctors obtained easier access to the records. Providers of photos could be assured their images were not being misused, without the many drawbacks that digital rights management (DRM) systems impose on those consuming the material. In a similar vein in which the growth of e-commerce Web sites led to the massive adoption of HTTPS, we envision that over time HTTPPA will be accepted by the larger Web community to meet the concerns of privacy and copyright violations on the Web.

Thesis Supervisor: Tim Berners-Lee
Title: Professor

Thesis Supervisor: Lalana Kagal
Title: Principal Research Scientist

Acknowledgments

This thesis is the culmination of an exciting seven year adventure at the Decentralized Information Group (DIG) at MIT. I am indebted to my supervisors Tim Berners-Lee and Lalana Kagal who had the vision to see the merits of an incoherent problem, and guide me in getting to the solution. Their guidance was indispensable for the successful completion of this thesis.

I was also very fortunate to have a fantastic thesis committee. The work in this thesis builds up on the ideas of Information Accountability put forward by many of my thesis committee members. Hal Abelson gave me lot of advice on how to frame the research question, Susan Landau gave me valuable advice on different perspectives of information accountability research, and Danny Weitzner has always urged me to posit this work in relation to the existing Web infrastructure.

This thesis would not have been possible without the many wonderful people I was fortunate to work with at DIG. The conversations, ideas, feedback and assistance from these individuals had lot of impact on this work. I thank K Krasnow Waterman, especially for her effort in trying to find real-world use cases for Accountable Systems. My fellow colleagues, both past and present at DIG: Fuming-Shih, Ilaria Liccardi, Daniela Miao, Sharon Paradesi, Evan Patton, Ian Jacobi, Mathew Cherian, and my friends outside MIT: Stephane Corlosquet and Puneet Kishor were always willing to listen to my crazy ideas and give suggestions and feedback.

I thank Sanjiva Weerawarana from WSO2 for getting me excited about Web technologies many years ago, and helping me with the decision to apply for grad school. Vishaka Nanayakkara, and other professors from my undergraduate institution University of Moratuwa also inspired me to pursue grad school, and I am forever indebted to their support.

Finally, I thank and dedicate this thesis my family: my parents Athula and Lakshmi - who have been my first and best mentors to science and engineering, and my husband Thilanka - who deserves lot of appreciation for his support during this PhD.

Contents

1	Introduction	19
1.1	Problems Addressed	19
1.1.1	Privacy Protection	20
1.1.2	Intellectual Property Rights Protection	22
1.2	State of the Art in Accountable Systems Research	23
1.2.1	Formal Theories for Accountability	24
1.2.2	Distributed Usage Control and Accountability	26
1.3	Contributions	27
1.4	The Definition for Accountable Systems	27
1.4.1	Accountable Systems:	27
1.5	The Approach for Accountable Systems	28
1.5.1	Characteristics of Accountable Systems	28
1.6	The Mechanism for Realizing Accountable Systems	29
1.7	Evaluation of Accountable Systems	30
1.8	Thesis Overview	31
1.8.1	Chapter 2: Early Experimentation	31
1.8.2	Chapter 3: An Accountable Architecture for the Web	31
1.8.3	Chapter 4: Use Case 1 - Privacy Enabling Transparent Systems	32
1.8.4	Chapter 5: Use Case 2 - Accountable Systems for Intellectual Property Rights Protection	32
1.8.5	Chapter 6: Related Work	33
1.8.6	Chapter 7: Conclusion	33

1.9	Summary	33
2	Early Experimentation	35
2.1	Understanding the Remix Culture on the Web	35
2.1.1	Methodology	36
2.1.2	Results	37
2.2	Measuring Policy Awareness on the Web	38
2.2.1	Policy Expression for Rights Management	38
2.2.2	Experiment for Measuring Attribution Violations on the Web	41
2.3	Client side Accountable Tools to Enable Appropriate Use of Content	46
2.4	Attribution License Validator	47
2.4.1	Design and Implementation:	48
2.4.2	Challenges and Limitations	49
2.5	Semantic Clipboard	49
2.5.1	Design and Implementation	51
2.6	License Server for Geo Spatial Data	52
2.6.1	Introduction	52
2.6.2	Design and Implementation	53
2.7	Summary	55
3	An Accountable Architecture for the Web	57
3.1	Motivation	57
3.2	Augmenting the Web with Accountability	58
3.2.1	Components of HTTPPA	59
3.2.2	HTTPPA Mechanism	61
3.3	Specifying Appropriate Use Through Usage Restrictions	61
3.3.1	Scenario Walkthrough	62
3.3.2	Usage Restriction Language	63
3.4	Usage Restrictions Management	64
3.4.1	Requesting a Resource with HTTP GET	65
3.4.2	Creating a Resource with HTTP PUT and POST	66

3.5	Provenance Tracking Auditing	66
3.5.1	The Provenance Tracking Network (PTN)	66
3.5.2	Accountability Logs	70
3.5.3	Accountability Checking	72
3.5.4	Performance Evaluation	75
3.6	Verification Service	78
3.7	Summary	79
4	Privacy Enabling Transparent Systems	81
4.1	Brief Introduction to EHR Systems	81
4.2	Scenario Walkthrough	83
4.3	Generating Provenance	84
4.3.1	Modeling the Data	84
4.3.2	Modeling the Events	85
4.3.3	Securing Logs in the PTN	87
4.4	Transparent Health	88
4.5	User Study on Transparent Health	88
4.5.1	Participant Profiles and Preliminary Questions	90
4.5.2	Creating the Health Profile	92
4.5.3	Setting Usage Restrictions	92
4.5.4	Simulating Information Flows	93
4.5.5	Auditing their Health Records	93
4.5.6	Reversing the Roles	94
4.5.7	Our Hypothesis and Supporting Anecdotes	94
4.6	Summary	94
5	Use Case 2 - Accountable Systems for Intellectual Property Rights Protection	97
5.1	Accountable Client: Accountable Image Macro Generator	98
5.1.1	Introduction	98
5.1.2	Implementation	99

5.1.3	Using the Tool	100
5.2	Complementary Accountable Server-Side Systems	106
5.2.1	Using the Accountable Photo Sharing Websites	106
5.3	Evaluation	108
5.3.1	Methodology	109
5.3.2	Results	112
5.4	Summary	115
6	Related Work	117
6.1	Privacy Protection	117
6.1.1	Platform for Privacy Preferences (P3P)	117
6.1.2	Notice and Choice Models	118
6.1.3	Transparency Enhancing Tools (TETs)	120
6.1.4	Privacy Protection in Electronic Healthcare Records (EHR) Systems	121
6.2	Intellectual Property Rights Protection	121
6.2.1	Making Attribution Easy	122
6.2.2	Allowing Fair Use	122
6.2.3	Reuse Detection	122
6.3	Provenance	123
6.3.1	Lineage of Data	123
6.3.2	Structured Logging	124
6.3.3	Semantic Web for Provenance	124
6.3.4	Inline Provenance using Metadata	125
6.4	Summary	125
7	Conclusion	127
7.1	Thesis Summary	127
7.2	Deployment Strategies	129
7.2.1	Consortiums to Maintain PTNs	129
7.2.2	Incentive Mechanisms	132

7.2.3	Policy Changes	134
7.3	Current Limitations	135
7.3.1	Scalability	135
7.3.2	Security and Privacy Concerns	136
7.3.3	Liability Concerns	138
7.3.4	Potential for PTN Abuse	138
7.3.5	Identity Requirements	138
7.4	Future Work	139
7.5	Conclusion	139
A	Summary of User Generated Content Websites Survey	141
B	License Lookup Table	143

List of Figures

1-1	Accountable Systems vs Rights Enforcement vs Free Culture	28
2-1	YouTube page displaying the license under a partner video.	37
2-2	Results from the experiment with the Attribution violations Rate and Precision Rates	43
2-3	The Design of the Validator with a sample output.	48
2-4	Semantic Clipboard Architecture and the User Interface	50
2-5	Components of the Geo-spatial Licensing Server	54
3-1	Building Blocks of Accountable Systems.	60
3-2	Sequence Diagram Resource Access in HTTPPA.	65
3-3	Sequence Diagram for Non-Existing Resources in HTTPPA.	67
3-4	Sequence Diagram for Existing Resources in HTTPPA.	67
3-5	The Provenance Tracking Network architecture	69
3-6	PTN and an Accountable Systems application interact through the PTN wrapper interface.	73
3-7	Auditing Usage Logs with the PTN	74
3-8	Average time it takes to complete authenticated get and put requests in a large deployment of a PTN consisting of 100 nodes during a time span of 24 hours	77

4-1	Provenance structure of data when a record was sent to another agent acting on another system. In this example, Doctor Dee refers a patient to Steven Special. For treatment purposes Steven needs the entire health record from General Hospital to be viewed.	85
4-2	Transparent Health: The complete health record of patients with the red 'Audit' button next to potentially sensitive information.	89
4-3	Audit Logs indicating how a sensitive data was used in Transparent Health. Depending on the usage restrictions set by the user, all the questionable usages appear on the left. The user can submit a question about the respective usage and subsequently flag the usage as a privacy violation.	90
4-4	Categorization of what users consider most useful in knowing if there is a mechanism to figure out privacy breaches of their sensitive information	91
5-1	Architecture of the Accountable Client Side Tool	99
5-2	Provenance Information Flow in the Accountable Client Side Tool . .	100
5-3	Accountable Client Side Tool on the Chrome Web Store	101
5-4	Options in the Accountable Client Side Tool	102
5-5	Using the Accountable Client Side Tool	102
5-6	Image Macro Generator Before Editing Image	103
5-7	Image Macro Generator After Editing Image	104
5-8	Client-side Audit Log for a Resource	105
5-9	Accountable Image Sharing Websites PhotoRM and ImageHare . . .	106
5-10	Upload Image in Accountable Photo Sharing Website	107
5-11	Uploaded Image with Metadata on the Accountable Photo Sharing Website	108
5-12	Audit Log for the Image Resource in PhotoRM	109
5-13	Perception of Study Participants as to the Potential Violation of Usage Restrictions given in the Audit Log	113
5-14	Reuse Behavior of Individuals with and without the accountability tool	115

7-1 Research Contributions to Computer Science 128

List of Tables

2.1	Attribution License Violations Rates of the Experiment Samples . . .	44
2.2	Precision Values After correcting for non 'self' attribution in the ex- periment samples	45
4.1	Secure Operations for Usage Logs on the PTN	87
A.1	User Generated Content Websites Survey	142
B.1	License Lookup Table	144

Chapter 1

Introduction

There is tremendous good from users sharing the right information with the right people in the right ways: scientists can use data in unexpected ways and discover groundbreaking results that can cure diseases; volunteers can crowdsource to find solutions that may take considerable time, effort and money otherwise. However, when information is taken out of context, or re-used in a manner that was not originally intended to be used for, there may be adverse consequences for the subject or the originator of the content. Access control mechanisms alone are not sufficient in these situations. In this thesis, I propose, implement, and evaluate a solution that supplements Access Control mechanisms in traditional Web Systems, but with Accountability.

In this chapter, I present the specific problems addressed with Accountable Systems on the Web, state of the art in Accountable Systems research, and an overview of the contributions in this thesis.

1.1 Problems Addressed

The Web has been designed as a decentralized information-sharing medium in which anyone, anywhere, can share any information simply by creating a document on a web server [1]. Many web based systems such as social networking websites, web-accessible health care records, personal tax report creation websites, personalized search and

other web-based information systems offer a variety of ways for netizens to engage socially, economically and intellectually with friends and strangers. These systems enable users to enter, use, and transfer sensitive information. There is an implicit trust by the users that the mechanics behind these web systems and other users will not misuse the personal information they provide to the system. In certain domains, such as healthcare or finance, the information usage is fairly complex and/or unpredictable that the user may not be completely aware about what is happening with their information, and the potential privacy implications of the information misuses. In this section we examine the types of problems on the Web that Accountable Systems provides an effective solution to.

1.1.1 Privacy Protection

Preserving privacy is a very difficult problem in the online age, as many social media outlets on the web provide an easy medium for an unprecedented level of information dissemination. Naive users divulge sensitive information without thinking much about their online privacy [2]. With the proliferation of social networking on the Web, the data people disclose on their social networking profiles has often been used against them [3]. A pure notice and choice model is also not adequate for privacy protection. Many companies offer information and services of any kind in exchange for users' personal data. Users who sign up for such services by disclosing their data such as email addresses and phone numbers often find themselves receiving unsolicited promotional offers, and worse, realize later that their data may have been sold to another party [4]. Most terms of service agreements, and privacy policies can be terminated by a bankruptcy court, a place where many of the startups who have lots of personal data from users will end up. Also 'User choice' is increasingly becoming a way for the websites to shift blame to users whenever a privacy breach happens. Users derive some benefits by being members of social networking websites and through interactions with popular websites. However the underlying benefits of these transactions to others remain hidden.

There are many privacy guidelines in effect today both nationally and internationally [5,

6]. Even though these guidelines have been around for many years there are no technical infrastructures that implements them. Most privacy and data protection systems utilize access control systems. In a pure access restriction system, those who obtain access to the data, legitimately or not, can use the data without restriction. Therefore, even when the information is within reasonable bounds of security with proper access control mechanisms, it can leak outside those boundaries violating the initial restrictions imposed on the data. On the other hand, there are situations where restricting access to data may cause more harm than good. For example, in a health care emergency, having access to the right information at the right time by the right people—regardless of them being authorized to access that information or not—might make the difference between life and death.

The scope of our work within privacy protection involves cases where the user has a reasonable expectation of the appropriate use of their sensitive data. For example, we often hand over our personal data to others, such as doctors, under the assumption that there is a mutual understanding that they will not divulge it to others unless it is related to diagnosis or treatment, store the sensitive data in encrypted databases, and always use secure means of communication. As the data subject, we (i.e. the patients) are interested in knowing if the doctors behaved in an accountable manner and that the sensitive data was used in an appropriate manner. Most of the recently reported privacy breaches happened not from malicious attacks on the systems, but rather from authorized insiders accessing and using private information in inappropriate manners [7]. Employees have abused their position to access sensitive information or created aliases to gather information from patients. For example, a licensed nurse practitioner accessed social security numbers of 919 patients during the period of September 2009 and October 2013 [8], a gynecologist secretly and illegally recorded images of more than 9000 patients [9], and various celebrities' private health records were snooped on [10, 11, 12, 13], and in some isolated cases co-workers, family members and neighbors were snooped on [14].

1.1.2 Intellectual Property Rights Protection

Reuse of intellectual property is one of the most important ways in which the Web is empowering the creation of knowledge and culture nowadays. For example, scholars like Manovich [15], Benkler [16] and Lessig [17] argue that this plays an essential role in the creation and development of today's culture due to the affordances that new technologies provide. Manovich points out that "remixing is practically a built-in feature of digital networked media universe", while Benkler positions this as a fundamental part of the way culture is produced and argues that if "we are to make this culture our own, render it legible, and make it into a new platform for our needs and conversations today, we must find a way to cut, paste, and remix present culture". Examples of content reuse, have existed for as long as creative work has existed. Musicians routinely use other songs and tunes in their compositions. Collage art is considered to be creative, and even original, although it is composed from many different sources.

The whole concept of Intellectual Property requires license information to be made available by publishers, and transmitted to the consumers via the appropriate channels. The 'Creative Community'—writers, artists, film-makers and so on—have the formal concepts of copyrights and licensing that make their expectations explicit. Engineering mechanisms such as Digital Rights Management (DRM) attempt to prevent others from copying and sharing these creative works. However, with such methods, access to content is completely sealed off, thus sounding a death knell to that generative culture that gives ever more creative things. Similarly, mash-ups are a peculiarly digital phenomenon of the Web age. They are entirely a product made possible by the portable, mixable and immediate nature of digital technology. A potential legal problem arises when more than one legally encumbered content or data stream is bound together with others in the form of a mash-up. The users of the original content should remain within the bounds of the permitted use of the components comprising the mash-up.

The current copyright law in the United States was passed in 1976, which, while

only 38 years ago, was more than a decade before the web came into existence. So there is arguably a need for copyright reform to keep pace with the technological changes [18]. Fair use, the legal doctrine that allows use of copyrighted content under certain conditions, is arguably the most confusing aspect of copyright law. The boundary between fair use and piracy seem to be thin if not permeable. A study on posts in community help forums has shown that there is poor literacy on the relevant laws, confusion on the moral and legal issues, conflicting advice, and suggestions to consult lawyers are in abundance [19]. There seem to be no clear answers to the following questions when users put their content online: what rights to that content do they grant? what terms are they agreeing to? can websites do anything they wish with the user generated content? Many users of the photo sharing site Instagram were seemingly unaware of answers to these questions. In December 2012, a provision in Instagram's Terms of Service allowing the use of photographs in connection with 'paid or sponsored content' gained some attention. Instagram later changed these terms after complaints from users that they do not wish their photos to be used in any advertising material [20]. This shows that those sharing creative works online may not always be aware of how their work can be used even though they have agreed to terms of service.

The scope of our work within intellectual property rights protection includes cases where we want to empower the content creators in the long tail of the content creation curve. We have designed, implemented and tested tools to make users aware of the copyright and licensing restrictions on content, and mechanisms to trace the usage of such content using Accountable Systems.

1.2 State of the Art in Accountable Systems Research

Taking a departure from traditional approach to information security through prevention, several researchers have put forward models to determine whether a rule has

been violated to ‘punish’ the violators in some fashion after the fact from audit logs. Many of these ideas supplement the traditional ideas of information security through prevention, and the work presented in this thesis is a systemization of some of these ideas. This section outlines some of the research fronts on accountable systems research starting from the formal definitions of accountability to more concrete models for implementing accountability.

1.2.1 Formal Theories for Accountability

In early work on accountability in Computer Science, Saltzer et al [21] have identified some deficiencies in ‘protection’ mechanisms in computer systems, and have put forward a solution that puts constraints on use of information after release by one program. The HalpernPearl framework of causality [22] gives formal definitions of both responsibility and blame to inform actions—such as punishment—taken in response to a policy violation.

Kusters et al. [23] have proposed a model-independent, formal definition of accountability that is focussed on systems to provide solid evidence of misbehavior when one occurs even if some parties misbehave in significant ways. This definition has been applied in three cryptographic tasks including contract-signing, voting and auctions. Their definition assumes there is an agent J , the *judge*, who is supposed to blame protocol participants in case of misbehavior. J can be a regular protocol participant or an external arbiter, and does not necessarily trust the other protocol participants. According to their definition J never blames protocol participants who are honest. Also, if the goal of the protocol is not met—due to the misbehavior of one or more protocol participants—then J blames these participants who misbehaved, or at least some of them. By contrast, the system we present, there is no hard and fast line defining rights and wrong, We do not model the social protocols exactly, and so do not precisely define what is “right” and what is “wrong” by the protocol, because in the scenarios we investigated, the ability to make that line flexible and a matter of judgement was essential. Also, rather than than a single judge J , it is the subject or the owner that may check the proper use.

Garg et al. [24] define accountability as it applied to privacy preservation in detecting policy violations, assigning blame and optimally managing risks stemming from privacy violations. They too have built up on the notion that *privacy is a right to appropriate flow of personal information* [25]. However, their focus has been mostly on the *expressivity* of practical privacy policies, and *enforceability* by checking whether traces satisfy policies expressed in the logic.

'Information Accountability' put forward by Weitzner et al [26], discusses an interesting departure from technology that enforces rules upfront. The focus there is to enable mechanisms for deterrence through public policy changes and system implementations in addition to precluding the possibility of violations. These should encourage compliance and maximize the possibility of accountability for violations without restricting the flow of information.

Feigenbaum et al. [27, 28] formalizes accountability with the following features: (1) unified treatment of scenarios in which accountability is enforced automatically and those in which it is enforced by a mediating authority and, (2) handle scenarios in which the parties who are held accountable can remain anonymous as well as those in which they must be identifiable by those whom they are accountable. Essential technical elements in this formalism include *event traces* and *utility functions*. In 'Systemizing Accountability', Feigenbaum et al. [29] explore three broad aspects of accountability: time, information, and action. The 'time' aspect considers five standard approaches to security violations: prevention, detection, evidence, judgement and punishment. The 'information' aspect considers the type(s) of credentials the system participants use, what constitutes evidence of compliance with or violation of a security policy, and who must have access to credentials and evidence for the system to function. The 'action' aspect examines the operations structures of accountability mechanisms, such as: are the actions decentralized or centralized? what action must be taken to deal with a violation?, etc.

Additionally, many computer scientists have developed logics with which to study accountability: Barth et al. [30] defined a logic for utility and privacy that they applied to models of business practices. Backes et al [31] used a protocol logic to

prove properties of contract-signing protocols including accountability protocols.

Within the purview of this work, I have used the concepts defined in ‘Information Accountability’ [26] and implemented a technical infrastructure to achieve an accountable web eco-system where deterrence plays a key role in appropriate use of information.

1.2.2 Distributed Usage Control and Accountability

The notion of accountability in distributed systems has been implemented in the PeerReview System [32]. This system maintains a tamper-evident record that provides non-repudiable evidence of all nodes’ actions. At each node in the network a detector module implementing the protocol will indicate either suspicion or certainty that another node is violating the protocol. It makes use of a tamper-evident log that each node maintains of its own interactions and can be obtained by other nodes as they need it. Every node that fails to acknowledge a message is eventually suspected of violating the protocol by every node that does follow the protocol.

Pretchner et al. in their work on ‘Distributed Usage Control’ propose enforcement of usage restrictions with regards to privacy sensitive personal data at various levels of abstractions in the operating system [33, 34, 35]. Distributed Usage Control Policies stipulate rights and duties concerning the future use of data. They present both a technology for data flow tracking within and across layers of abstraction and within and across single machines, declassifications based on data quantities, provenance tracking, and ways to protect the infrastructure itself.

The work presented in this thesis implements technology similar to the PeerReview System, but with increased decentralization where anybody anywhere can participate in the network of trusted servers that is used in the arbitration of appropriate use of data. We have also given thoughtful consideration to Distributed Usage Control, especially in modeling the temporal, spatial, quantitative, technical and organizational constraints and purposes in modeling the usage restrictions. In trusted systems environments such as distributed usage control systems, data can be moved from one application to another only in ways that are implemented by the application, and the

only data manipulations permitted are those that are coded into the app. This design limits flexibility, but may lead to a more secure and predictable user experience.

1.3 Contributions

Rather than concealing information to be overly cautious of privacy, or intellectual property protection, the work presented in this thesis seeks to enforce fair use at the point of consumption. It provides transparency through auditing, enabling the data subject or owner to determine any inappropriate use of her information after the fact. The core contributions of this thesis can be summarized as follows:

- One of the first **comprehensive technical Accountable Systems infrastructures** to enable appropriate use of information, and determine inappropriate uses of information on the Web
- Solution for Privacy Violations in Web based Electronic Healthcare Systems
- Solution for Intellectual Property Violations

1.4 The Definition for Accountable Systems

As discussed in Section 1.2.1, different researchers have explicitly or implicitly used the term ‘Accountable Systems’ to mean different things. The word ‘accountability’ might signal a definition along the lines of taking responsibility in the face of fraud or malice. In other contexts it might signal assigning blame to an individual. To avoid any ambiguity in the meaning of Accountable Systems, we define Accountable Systems as follows.

1.4.1 Accountable Systems:

An accountable system is one in which it is possible to tell whether each use of data is compliant with a given set of rules or the usage restrictions

indicated by the user, and, when a violation occurs has occurred, it is possible to determine the source of that violation.

If use of information in web based systems can be displayed on a spectrum as shown in Figure 1-1, Accountable Systems fall somewhere in-between the rights enforcement and free culture where there is just the right balance benefiting both the information or content provider and the consumer.



Figure 1-1: Accountable Systems vs Rights Enforcement vs Free Culture

1.5 The Approach for Accountable Systems

The Web provides a medium where users can engage in activities anonymously or pseudonymously, therefore making it difficult to have the identity of any perpetrators known. Also, there are no means of determining the provenance of a piece of information as it gets shared and reused across the Web. Accepted or permitted use is usually specified in a superficial manner where the meaning of ‘use’ is specific to just one website. Considering these current limitations of the Web, we suggest the following general approach for driving accountability on the Web.

1.5.1 Characteristics of Accountable Systems

- Agents engaging in accountable systems should have a **unique identity**. The desired property of the identity mechanism is that it should be addressed in the global namespace without any collisions. There can also be different authentication mechanisms as long as they can all be unified to a single identity profile.

- Each piece of information may have a **usage restrictions** attached with it. The desired property of the usage restriction is that it is declared in a machine readable format and can be represented in a dereferenceable URI that gives the interpretation of the rule specifying the usage restriction.
- Information can live in, be shared on, and used on multiple systems that all speak the same protocol. These systems can be independent with respect to the data storage mechanism, the processes and the agents that act on the data.

1.6 The Mechanism for Realizing Accountable Systems

The technical contributions of this thesis are twofold: (1) technologies that enable appropriate use of data, and (2) technologies that help determine inappropriate use of data through provenance. A comprehensive accountable system as defined in Section 1.4 will have both these aspects, and in this thesis systems utilizing either or both these aspects are demonstrated.

At the core of the mechanism for realizing Accountable Systems is HTTPA (Hyper Text Transfer Protocol with Accountability). HTTPA is an end-to-end protocol which extends HTTP [36] to provide appropriate use negotiation for accountability. Therefore it needs both servers and clients (such as your web browser) all to agree on the protocol terms. When a client requests a resource in HTTPA, for example, a picture, or a patient's medical record, from a server, the server will send over the usage restrictions. An example usage restriction could be: 'do not make any derivatives from this resource', or 'do not share this resource'. As soon as the client sends the agreement, the server logs the transaction in a special purpose network of servers and sends the resource to the client. In the systems we present, the client software passes on the restrictions to the user, but does not, in general, enforce them. However, the

usage actions, such as making any derivatives of the resource data, are logged in the special purpose network of servers. The owner of the resource can consult the special purpose network of servers and request an audit log for that resource at a later point and determine if any usage restriction violations have taken place.

1.7 Evaluation of Accountable Systems

We argue that in systems that support health care decisions or military information systems where the safety of an individual or a community is at risk, foregoing access control mechanisms and getting to the correct information fast through accountable mechanisms is arguably the better alternative compared to upfront preventive measures. In these cases, the data subject will still need some assurance that their sensitive information was not misused by anyone who was authorized. Even in cases such as referrals to other medical organizations, the patient will be interested in knowing how her information moved, who had access to it, where the information was used in, etc. To model this scenario using an Accountable Systems implementation, we developed a prototype Electronic Health Care Records System, where different organizational units in the healthcare scenario were modeled to have their own data stores, processes, and users. In this prototype, the patients were given a transparent view as to how their sensitive information was used.

We conducted another evaluation of Accountable Systems in intellectual property rights domain, where we created prototype photo sharing websites that anybody can login using an OAuth provider such as Google, upload pictures with a Creative Commons license and share them on the Web as they fit without any enforcement. A complementary client side tool was made available to edit images that are discovered while browsing the Web, and share them on popular websites as well as on two HTTPPA websites. If the end-to-end Accountable System comprising of the HTTPPA websites and the Accountable client side tools were used, the content providers can determine how their content was used, and the content consumers are able to ascertain where the content came from, what usage restrictions it has, etc. as HTTPPA communicates

the usage restrictions per resource, and it is possible to ascertain the origin and reuses of the resource.

1.8 Thesis Overview

Each subsequent chapter in this document highlights some of the contributions in this thesis. These chapters are summarized here.

1.8.1 Chapter 2: Early Experimentation

This chapter includes some early experiments I conducted to gather requirements for Accountable Systems. These include an experiment on Creative Commons license violations on Flickr images reused in many relatively high volume blogs, and a survey to determine the level of support in online tools and websites for sharing and reuse on several user-generated content websites. Using the lessons learnt in our studies, I created a validation tool and a clipboard for images, and a license server for geo-spatial data to solve the specific reuse issues in the corresponding application domains. Unlike previous work on data reuse (i.e. Digital Rights Management, Distributed Usage Control), there is no enforcement mechanism in these tools. Rather, these tools advise users on the proper usage of content, and follow the approach advised by Creative Commons. These tools provided early exploratory feedback on the requirements for a comprehensive Accountable Systems Infrastructure on the Web.

1.8.2 Chapter 3: An Accountable Architecture for the Web

Using the lessons learned from the early experimentations, I developed technologies to provide accountability on the Web. These functionalities are embodied in a protocol called HTTPPA that makes use of a distributed network of trusted servers called the Provenance Tracking Network (PTN). HTTPPA extends HTTP with principles of accountability in order to enable appropriate use of information on the Web. This chapter describes the details of HTTPPA, and the methodology for determining inap-

propriate use of data through HTTPPA.

1.8.3 Chapter 4: Use Case 1 - Privacy Enabling Transparent Systems

We implemented a reference Accountable Systems implementation using HTTPPA called ‘Transparent Health’, a prototype Electronic Healthcare Records System, in which a user can mark some data items as sensitive and later audit them. This system evaluated the privacy axis of the work presented in this thesis. We conducted a user study to evaluate this system, and found that many participants preferred using a system where they can ‘audit’ their personal health data. The details of the implementation of Transparent Health and the evaluation of it are described in this chapter.

1.8.4 Chapter 5: Use Case 2 - Accountable Systems for Intellectual Property Rights Protection

Implementing usable client side technologies for Accountable Systems is of utmost importance to get the technology off the ground. We illustrate this with a simple photo editing tool on the browser that implements HTTPPA. PhotoRM and Imagehare are reference photo share websites that implement HTTPPA, and the Accountable Meme Generator is the client side implementation of HTTPPA. The tool understands the protocol and provides user cues as to what the corresponding usage restriction for a resource is, and any derivations of it. We used this set up to evaluate the intellectual property rights axis of this thesis and conducted another user study with more than 150 participants, where a majority of them indicated that they would like to see such a feature on the Web.

1.8.5 Chapter 6: Related Work

This chapter presents some important related work in privacy, intellectual property rights violations and in provenance management systems on the Web. I compare and contrast these works with our solutions.

1.8.6 Chapter 7: Conclusion

This chapter gives a summary of the thesis, limitations of the technology, some deployment strategies that are required to move this project from research to the real-world and proposals for future work.

1.9 Summary

If we were to make data strictly private with very strict access control policies, we could be throwing the baby out with the bath water, and making a world in which nobody will benefit from the wealth of information available on the Web. Similarly, in the intellectual property rights domain, artists and content creators should be able to freely share their content without worrying about someone stealing their creations, and expect to be attributed and credited for their work appropriately. This thesis addresses these problems, i.e. enabling appropriate use of and determining inappropriate use of information, and provides several technical solutions.

I present the design and implementation of a system that has support for ‘break glass’ scenarios where information can be accessed when necessary, but logs all activity for compliance checks afterwards in order to *determine* inappropriate uses of information, and subsequently enable tracking usage of content. In addition, this thesis presents several tools that can be used to *enable* appropriate use of information, by making it easy for the user to do the right thing, instead of making it difficult to do the wrong thing.

In order to understand the scale of copyright violations on the Web, I conducted several experiments and user studies. Realizing the issues that are currently on the

Web with regards to content reuse on the Web from these studies, I developed several user facing tools to encourage the appropriate use of copyrighted media. These studies and the tools are described in the next chapter.

Chapter 2

Early Experimentation

In order to understand the scale of the problem of inappropriate use of content, I conducted several studies to gauge the level of inappropriate use of content on the Web. Learning from the results of the studies, I then implemented several tools that can be used to enable policy awareness when reusing such content. Even though both tools were limited to image reuse, it can be easily extended to support other types of media. These client side tools motivated the need to have a comprehensive architecture for achieving accountability, as awareness of policies or usage restrictions are not enough for a robust end-to-end Accountable System. This chapter describes in detail the studies on the reuse culture of the Web, violations estimate of Creative Commons licenses and few tools that enable the user to appropriately reuse content.

2.1 Understanding the Remix Culture on the Web¹

The requirement for attribution is a form of social compensation for reusing one's work. While mentioning one's name, homepage or any other kind of identifier when attributing draws attention to an individual, other forms of '*attention mechanisms*' can also be implemented. For example, a content creator can obligate the users of her works to give monetary compensation or require that they include certain ad links in

¹The work described in this section is published at the Web Science conference in 2010 in the paper titled 'Remix culture on the web: a survey of content reuse on different user-generated content websites' [37].

the attribution HTML or give attribution in an entirely arbitrary manner. A similar study was done by Hill et al [38] on the Scratch online community² about the remix behavior of kids in a creative learning environment. The goal in this survey however was to elicit any opportunities for remix behavior through the functions provided in several popular user generated content websites. I also wanted to better understand what is lacking in the current web ecosystem with the intention of developing tools that can provide solutions to those missing pieces.

I examined how content reuse is supported in a wide range of user-generated content websites, including photo sharing, video sharing, audio sharing, scientific data sharing, social networking, etc. to understand what is lacking in the current web eco-system to enable accountability in accessing, repurposing and sharing content. In particular, I analyzed the ways in which these systems support some the following forms of content reuse: (1) Licensed content brought in from external sources. (2) Content uploaded to the website and given a license of the user's liking. (3) Licensed content from the website reused elsewhere. (4) Licensed content reused within the website.

2.1.1 Methodology

I chose to study six major categories of popular user-generated content websites. Namely: video sharing, photo sharing, audio sharing, micro-blogging, social networking, and scientific data sharing websites. In analyzing the websites for this survey, I tried to elicit answers for questions such as: can licensed content be brought from an external source?, what are the licenses supported by these websites?, how easy is it to find licensed content in the website?, how easy is it to legally remix content using the features available in the website?, how effective are the mechanisms for credit giving? are they automatic? or is the user required to give credit manually?.

Since a healthy remix culture requires crediting the original providers, I also looked at how well some of the major content sharing websites support the remix culture by

²The Scratch website makes it easy to reuse, remix and share stories and animations. The website can be accessed at <http://scratch.mit.edu/>.

empowering people to build on the work of those creators who want their work to be reused; and made it easier to understand what are the rights of authors and the licenses associated with their work.

2.1.2 Results

Majority of the websites analyzed had insufficient support for these different types of content reuse scenarios, and were merely acting as content distribution portals rather than platforms that support a culture of remixing and sharing. Some of the websites were found to have support for different licenses but most websites lack adequate functionality for finding licensed content, remixing and attributing sources for remixes. For example, YouTube has support for Creative Commons licensing for special partners (Figure 2-1), whereas websites such as Flickr has support for Creative Commons licenses by default and they can be customized.

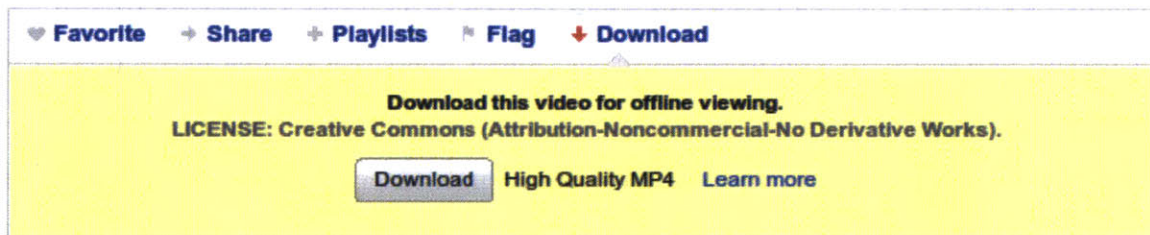


Figure 2-1: YouTube page displaying the license under a partner video.

I found deficient support mechanisms for entering, presenting and linking licensing and attribution information in both human and machine readable formats. The findings are summarized in Table A.1 in Appendix A .

In order to address the issues identified here, I propose simple design interventions that include:

1. Letting content creators choose the license for their work and display this license in human and machine readable form.
2. Allowing content uploaders to give credit to the sources of their work by providing hyperlinks and metadata of such sources.

3. Displaying provenance information that display the tree of derivative work of some content as well as its antecedent work.
4. Enabling remixing of content from a different system.
5. Giving people the tools to easily embed and remix content in a way that follows the license chosen by its creator.

2.2 Measuring Policy Awareness on the Web³

2.2.1 Policy Expression for Rights Management

Policies in general are pervasive in web applications. They play a crucial role in enhancing security, privacy and usability of the services offered on the Web [40]. A license is an instrument that conveys a right, accompanied by a promise, by the grantor to not sue the grantee if that right is exercised. In the context of digital files, a license describes the conditions of usage under which those files may be used. A license on a digital file can exist whether or not there are any corresponding users of that file. A user would have to abide by the license that covers the usage of that file, and if any of the conditions of usage described in that license are violated, then the user would have to cease using that file. Licenses are covered by copyright laws.

Policies governing the reuse of digital content on the Web can take several forms. It can be upfront enforcement mechanisms such as *Digital Rights Management* (DRM) approaches, or rights expression mechanisms such as *Creative Commons* licenses where users are given the freedom to reuse content, subject to several restrictions and conditions.

Digital Rights Management: The cryptography community has much work on identifying illicit flows of files. Such work has included watermarking, and fingerprinting [41, 42], in which a key that links to the identity of the person to whom the publisher sold the copy is embedded in the work. The distribution and usage of

³This experiment that outlines the need for appropriate use of data through license awareness is included in the publication titled 'Policy Aware Content Reuse on the Web' published at the International Semantic Web Conference in 2009 [39].

copyrighted content is often controlled by up-front policy enforcement. These systems usually restrict access to the content, or prevent the content from being used within certain applications. The core concept in DRM is the use of digital licenses, which grant certain rights to the user. These rights are mainly usage rules that are defined by a range of criteria, such as frequency of access, expiration date, restriction to transfer to another playback device, etc. An example of a DRM enforcement would be a DRM enabled playback device not playing a DRM controlled media transferred from another playback device, or not playing the media after the rental period has ended. DRM prevents end-users from using content in a manner inconsistent with its creator's wishes. The license describing these use-conditions typically accompanies the content as its metadata, and this technology is generally considered to be prohibitive.

Creative Commons (CC): In contrast, CC has been striving to provide a simple, uniform, and understandable set of licenses that content creators can issue their content under to enable reuse with much less restrictions. With almost a hundred different license available for a creator of digital data, what is out there is a babel of licenses [43], and CC makes our task easier by clearing up the minefield of licenses. Through an innovative license chooser that makes the job of choosing the appropriate license easy even for a non-initiate, and through a choice of less than half a dozen possible licenses, CC presents an innovative solution to an otherwise confusing problem.

For the experiment I chose CC licenses because they can be expressed semantically, widely deployed on a range of media, and have a large community base. CC licenses provide a very clear and a widely accepted rights expression language implemented using Semantic Web technologies [44]. These licenses are both machine readable and human readable, and clearly indicate to a person, who wishes to reuse content, exactly how it should be used by expressing the accepted use, permissions, and restrictions of the content. Popular search engines, including Google, Bing, and even sites such as Flickr, blip.tv, OWL Music Search and SpinXpress, have advanced search options to find CC licensed content on the Web [45, 46, 47, 48, 49].

Types of CC Licenses

Often, people tend to post their content with the understanding that it will be quoted, copied, and reused. Further, they may wish that their work only be used with attribution, used only for non-commercial use, distributed with a similar license or will be allowed in other free culture media. To allow these use restrictions CC has composed four distinct license types: *BY* (attribution), *NC* (non-commercial), *ND* (no-derivatives) and *SA* (share-alike) that can be used in combinations that best reflect the content creator's rights.

Generating CC Licenses

In order to generate the license in XHTML easily, CC offers a license chooser that is hosted at <http://creativecommons.org/license>. With some user input about the work that is being licensed, the license chooser generates a snippet of XHTML that contains the RDFa [50] to be included when the content is published on the Web.

Creative Commons Rights Expression Language

Content creators have the flexibility to express their licensing requirements using the Creative Commons Rights Expression Language (*ccREL*)⁴ [44] and are not forced into choosing a pre-defined license for their works. Also, they are free to extend licenses to meet their own requirements. *ccREL* allows a publisher of a work to give additional permissions beyond those specified in the CC license with the use of the *cc:morePermissions* property to reference commercial licensing brokers or any other license deed, and *dc:source* to reference parent works.

Exposing CC Licenses

Typically there are two ways in which metadata about licenses can be exposed.

- (1) Through APIs: For example, Flickr allows users to specify the license associated with their images. These license information can then be queried through the

⁴*ccREL* is the standard recommended by the Creative Commons for machine readable expressions of the meaning of a particular license.

Flickr API. This method is not very interoperable as API specific data wrappers have to be written for each service.

- (2) Through Resource Description Framework in Attributes (RDFa) [50]: CC licenses can be expressed in machine readable form using RDFa. The content creator and consumer can use RDFa for rights expression and compliance respectively. RDFa allows machine understandable semantics to be embedded in HTML.

2.2.2 Experiment for Measuring Attribution Violations on the Web

Unless a particular piece of content on the Web has some strict access control policies, I suspected that most users do not feel the need to check for the license it is under and be license compliant. To verify this hypothesis I conducted an experiment to assess the level of attribution license violations.

The goal of the experiment was to obtain an estimation for the level of CC *attribution license violations* on the Web using Flickr images.

Why Flickr? I chose Flickr because the site has over 300 million Creative Commons Licensed images. Thus it provided a large sample base for our experiment.

Why Attribution License Violations? The requirement for attribution is a form of social compensation for reusing one's work. While mentioning one's name, homepage or some other kind of identifier when attributing, draws attention to an individual. However, other forms of '*attention mechanisms*' can also be implemented. For example, a content creator can obligate the users of her works to give monetary compensation or require that they include certain ad links in the attribution HTML or give attribution in an entirely arbitrary manner.

Sample Collection for the Experiment

The samples were obtained using clusters of web pages indexed during a particular time frame in a blog indexing software called the Technorati blog indexer. I limited the number of web pages to around 70, and the number of images to less

than 500, so that I could do a manual inspection to see if there are any false positives, false negatives and/or any other errors. This also enabled us to check if the different samples contained the same web pages. I found that the correlation among the samples was minimal. I retrieved results for web pages linking to Flickr server farm URIs that have this particular format: `http://farm<farm-id>.static.flickr.com/<server-id>/<id>_<secret>.(jpg|gif|png)` using the query functions available in the Technorati blog indexer.

I made sure that the samples were independent of each other and the correlation among the samples were low by running the experiment three times with two weeks between each trial. This is because the *Authority Rank* given to a web page by Technorati, and hence the results returned from the query functions dynamically changes as new content gets created.

Criteria for Checking Policy Awareness

Checking policy awareness entailed checking for attribution information when the images are reused in the blogs.

However, Flickr does not use ccREL. Therefore, Flickr users do not have that much flexibility in specifying their own *attributionURL* or the *attributionName* as specified in ccREL. However, it is considered good practice to give attribution by linking to the Flickr user profile or by giving the Flickr user name (which could be interpreted as the *attributionURL* and the *attributionName* respectively), or at least, point to the original source of the image [51].

Therefore, the criteria for checking attribution consist of looking for the *attributionURL* or the *attributionName* or any *source citations* within a reasonable level of scoping from where the image is embedded in the Document Object Model (DOM) of the corresponding Web page. The results from the 3 trials are given in Fig 2-2. On the left, there are the sample screenshots of the results from the experiment. On the right, attribution violations rate and precision obtained after correcting for self-attribution that can be counted as false positives. These results have misattribution and non-attribution rates ranging from 78% to 94% signaling that there is a strong

need to promote license or policy awareness among users of content with the hope that they will follow the terms specified in the licenses.

Results

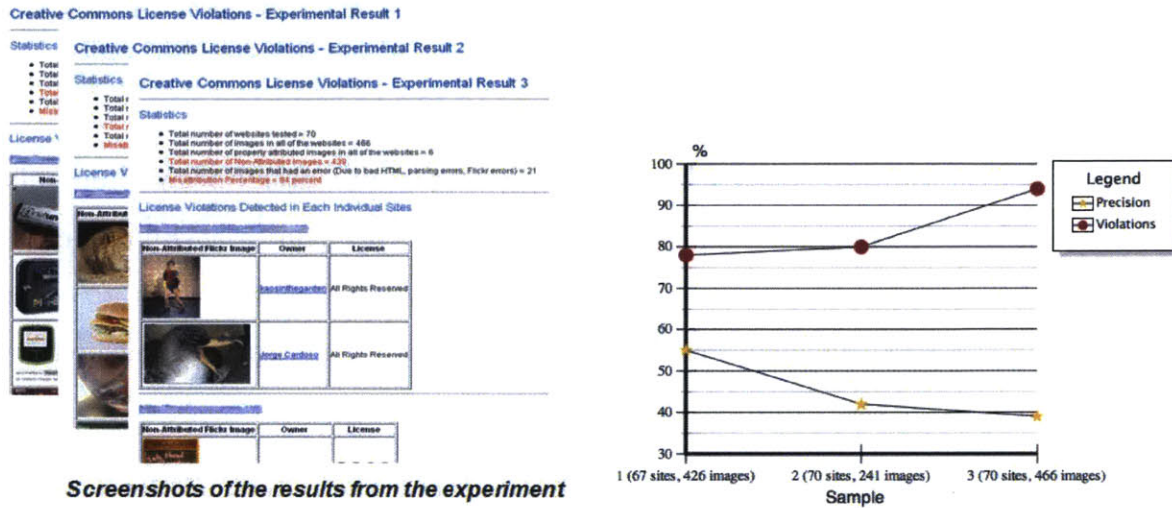


Figure 2-2: Results from the experiment with the Attribution violations Rate and Precision Rates

The entire result set includes the total number of Web pages tested, number of images in all of those Web pages, number of properly attributed images, number of misattributed or non-attributed images, and the number of instances that led to an error due to parsing errors resulting from bad HTML. Using these values, the percentages of misattribution and non-attribution for each sample were calculated as shown in Table 2.1. As shown in the left half of Fig 2-2 for each of the offending sites, the interface gives the non-attributed or the mis-attributed image, the original owner of the image and the license it is under.

We have only explored one domain of content, specifically images. However, there are billions of videos uploaded on YouTube, and potentially countless number of documents on the web, which have various types of licenses applied. Thus a solution of this nature which detects CC license violations based on the metadata of other types of free-floating Web media will be very useful.

	Sample 1	Sample 2	Sample 3
Websites	67	70	70
Total images	426	241	466
Non-attributed images	312	192	435
Mis-attributed images	28	0	3
Total violations %	78%	80%	94%

Table 2.1: Attribution License Violations Rates of the Experiment Samples

Limitations of the Experiment

(1) Results include cases where the users have not attributed themselves.

For example, consider the case where a user uploads her photos on Flickr, and uses those photos in one of her own web pages without attributing herself. As the copyright holder of the work, she might be of the opinion that she can do pretty much whatever she wants with those. However the CC BY license deed states:

“If You Distribute you must keep intact all copyright notices for the Work and provide the name of the Original Author...” [52]

While this is open for interpretation, I believe that self-attribution is important because otherwise it might set a precedent for the violation of user’s own rights in the long run. Therefore, I counted cases of non ‘self’ attribution as cases of non attribution.

However, I thought it would be interesting to discount cases of ‘self’ attribution as *false positives* and obtain the precision. Since it was hard to make a correlation between the Flickr photo owner and the page owner algorithmically, I manually inspected the samples to see whether the misattributed images were actually from the user or not, and flagged the ones which are definitely from the original user as *false positives* in the results set. After this correction, I found the precision rate of the experiment to be between 55% to 40%. The results are summarized in Table 2.2.

(2) Low adoption of ccREL and Attribution Scoping. I found out that a majority of the web pages examined in this experiment have not used ccREL in marking up attribution. Therefore, I used a heuristic to check for the existence of attribution in the pages used in the trials. This heuristic includes the *attributionName*

	Sample 1	Sample 2	Sample 3
Total images	426	241	466
Non 'self' attribution correction	183	113	268
Precision	55%	42%	94%

Table 2.2: Precision Values After correcting for non 'self' attribution in the experiment samples

constructed from the Flickr user name, or the *attributionURL* constructed from the Flickr user profile URI, or the original source document's URI. I expected to find the attribution information in the parent of the DOM element or in one of the neighboring DOM elements. This can be visually correlated to finding the attribution information immediately after the content being reused. However, since there is no strict definition from CC as to how attribution should be scoped, someone could also attribute the original content creator somewhere else in the document. Also considering that it is possible the user intended to include more than one image from the same original content creator, and by mistake failed to attribute some images, while correctly attributing all the others, I only checked attribution information within the neighboring DOM elements, and not at the document level.

(3) Blog Aggregators such as Tumble-logs cutting down the text and favoring short form, mixed media posts over long editorial posts: Use of such blog aggregators (for example `tumblr.com`) is another problem in getting an accurate assessment of attribution license violations. For example, in a blog post where a photo was reused, the original owner of the photograph may have been duly attributed. But when the *tumble-log* pulls in the feed from that post in the original web page and presents the aggregated content, the attribution details may be left out. This problem is difficult to circumvent because there is no standard that specifies how aggregation should take license and attribution details into consideration.

(4) Experiment Results Limited to Images: We have only explored one domain of content, specifically images. However, there are billions of videos uploaded on YouTube, and potentially countless number of documents on the web, which have various types of licenses applied. Thus a solution of this nature which detects CC

license violations based on the metadata of other types of free-floating Web media will be very useful.

2.3 Client side Accountable Tools to Enable Appropriate Use of Content

As our license violations experiment indicated, there is a strong lack of awareness of licensing terms among content users. This raises the question as to whether machine readable licenses are actually working. Perhaps more effort is needed to bring these technologies to the masses, and more tools are needed to bridge the gap between the license-aware and the license-unaware. There should also be methods to find out license violations when users are not cooperating. Therefore, even with these human-friendly CC licenses and the tools to support license discovery, license violations occur due to many reasons: Users may be ignorant as to what each of the licenses mean, or forget or be too lazy to check the license terms, or give an incorrect license which violates the original content creator's intention, or intentionally ignore the CC-license given to an original work in their own interests.

More generally, we identify four problems with the typical methods of conveying a license:

1. **Separation:** Even if metadata are created, they can and do get easily separated from the data.
2. **Interoperability:** When two or more datasets or contents are reused together, the user has to figure out what use permissions the newly created dataset would inherit. While for most metadata, the solution is simple accretion, for licenses this can get tricky because different licenses can be at odds with each other making the datasets incompatible.
3. **Generativeness:** Building upon the interoperability problem, we use the term generativeness to describe the problem pertaining to generation, the ability to produce a new license easily.

First, if we embed the license within the data, much like EXIF/IPTC metadata in photographs, it solves the separation problem. Second, if we utilize a standard, rights expression language such as ccREL, we solve the interoperability problem. And, finally, we programmatically compute the new license for a data set created by mashing-up two or more creative content or data sets by using the license composition given in Appendix B, thereby solving the generative problem. Generating a license describing the use-conditions automatically is not only easy for the user, but it is also less prone to errors.

Therefore, it is important that we have tools and techniques to make users aware of policies, usage restrictions and licenses that they must follow while making the process of being policy-compliant as painless as possible for the user, and make it difficult for someone to become license in-compliant either deliberately or by mistake. In the subsequent sections I outline two tools that

2.4 Attribution License Validator⁵

When someone aggregates content from many different sources, it is inevitable that some attribution details may be forgotten. While tools such as <http://validator.creativecommons.org> provided by the CC detects the embedded licenses in a page and gives information about the license, it cannot be used to detect a license ‘violation’ and obtain helpful hints as to how to make the user’s composite document compliant with the original source licenses. The Attribution License Violations Validator is designed to check whether the user has properly cited the source by giving the due attribution to the original content creator. In order to make sure that no CC license terms of the user are violated, the author can run the CC License Violations Validator and see if some sources have been left out or whether some have been misattributed to be accountable for her actions.

⁵The Flickr Creative Commons Attributions License Validator is available at <http://dig.csail.mit.edu/FlickrCC/validator.cgi>

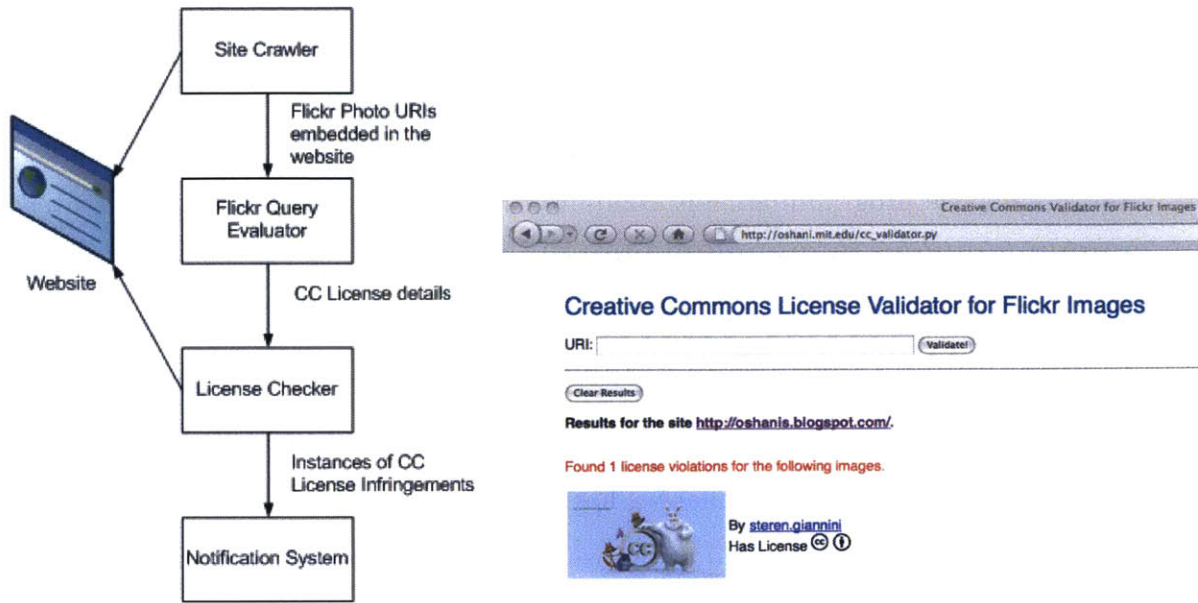


Figure 2-3: The Design of the Validator with a sample output.

2.4.1 Design and Implementation:

The tool has four major components as shown in the left half of Fig 2-3. On the right hand side is the output from the Validator showing the image that was not attributed properly, who the image belongs to and what license it is under. Once the user gives the URI where the composite work can be found, the site crawler will search for all the links embedded in the given web page and extract any embedded Flickr photos. From each of these Flickr photo URIs, it is possible to glean the Flickr photo id. Using this photo id, all the information related to the photo is obtained by calling several methods in the Flickr API. This information includes the original creator’s Flickr user account id, name, and CC license information pertaining to the photo, etc. Based on the license information of the Flickr photo, the tool checks for the attribution information that can be either the *attributionName*, *attributionURL*, source URI or any combination of those within a reasonable scoping in the containing DOM element in which the image was embedded. The ‘reasonable scoping’ in this case, is taken to be within the parent or the sibling nodes of the element that has the embedded image. If such information is missing, the user is presented with the details

of the original content creator's name, the image along with its URI, and the license it is under, enabling the user to compose the HTML required to properly attribute the sources used.

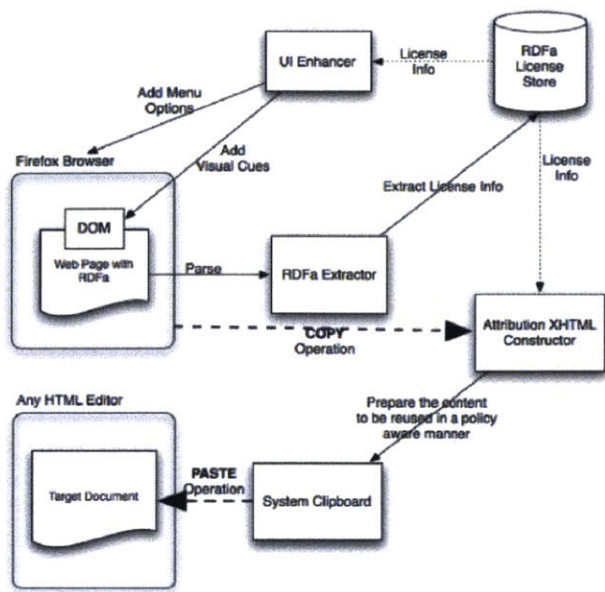
2.4.2 Challenges and Limitations

The license violations detection can only work if the image URI is actually linked from the Flickr site. Another complication is that a Flickr user can upload and assign CC licenses regardless of that user having the actual rights to do so. In other words, if someone uploads a copyrighted photo from *Getty Images* and assigns a CC license on Flickr, and an innocent user downloads and uses this photo, then that user will be violating the copyright law inadvertently. Therefore, we need to have some capability to track provenance of image data, and be able to identify whether a particular image has been used elsewhere in a manner that violates the original license terms. One of the major assumptions we have made in developing this tool is that attribution is specified within the parent node or the sibling nodes of the containing image element. Otherwise we classify it an instance of non-attribution. This assumption works in practice and appears to be the most logical thing to do. However, since there is no standard agreement as to what the correct scoping for attribution is, this assumption can lead to a wrong validation result. The solution to this problem is two-fold. (1) CC should give a guideline as to what the correct scoping of attribution should be relative to the content that is attributed. (2) Flickr (or any other such service) should expose the license metadata as RDF, instead of providing an API to query with. Exposing license metadata as RDF is preferred as it enables data interoperability and relieves the tool authors from having to write data wrappers for each service.

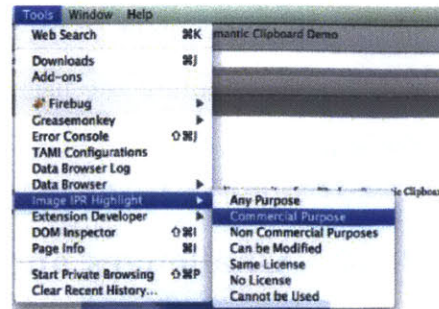
2.5 Semantic Clipboard⁶

Realizing the shortcomings of many tools available that enable users to be license compliant, I developed the Semantic Clipboard as a Firefox Web browser based tool

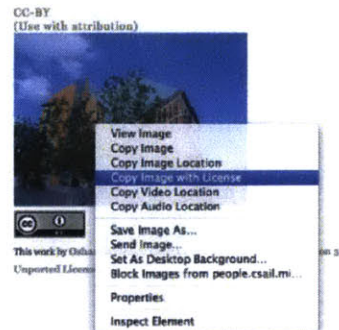
⁶Semantic Clipboard is available at <http://dig.csail.mit.edu/2009/Clipboard/>



Design of the Semantic Clipboard



Firefox Menu



Context Menu on an Image

Figure 2-4: Semantic Clipboard Architecture and the User Interface

that can be installed as a Firefox extension [53]. For example, license aware Mozilla Firefox extensions such as MozCC [54] provides a specialized interface for displaying CC licenses, where the user receives visual cues when a page with RDFa metadata is encountered. This includes the display of specific CC branded icons in the browser status bar when the metadata indicates the presence of a CC license. However, this software does not offer the capability to copy the license attribution HTML as in the Semantic Clipboard that we have developed. Additionally, there are several tools that can be used to automatically embed the license metadata from Flickr. Applications such as ThinkFree, a Web based commercial office suite [55], and the open source counterpart of it, the “Flickr image reuse for OpenOffice.org” [56] are examples of such applications. These applications allow the user to directly pick an image from the Flickr Web site and automatically inject the license metadata with it into a document in the corresponding *office suite*. A severe limitation of this approach is that they only support Flickr images. The Semantic Clipboard can be used to copy any image to any target document along with the license as long as the license metadata is expressed in

RDFa. Attributor [57] and PicScout [58], both commercial services, carry out online image infringement detections for their customers. They then offer to send notices to the offending web sites notifying link requests, offers for license, requests for removal or shares of the advertisement revenue from the offending pages. The problem with these services is that they penalize the infringers, rather than encouraging them to do the right thing upfront [59]. Also, these services are not free, which bars many content creators who wish to use such services to find license violations of their content from using the service.

2.5.1 Design and Implementation

The Semantic Clipboard was actually inspired from the work done on ‘HTML Documents with Inline, Policy-Aware Provenance’ [60] by Harvey Jones. Jones developed a document format that can represent information about the sources of its content, a method of excerpting from these documents that allows programs to trace the excerpt back to the source, a CC reasoning engine which calculates the appropriate license for the composite document, and a bookmarklet that uses all these components to recommend permissible licenses. But this tool requires all the source documents to be annotated with a special document fragment ontology, and the *Paste Operation* is limited to inserting copied HTML in the top level of the document only, i.e. it does not allow copying inside existing document fragments. The Semantic Clipboard addresses these issues by eliminating the reliance of an external document fragment ontology and utilizing the operating system’s clipboard to paste the image with the associated license metadata in HTML. The only requirement for the Semantic Clipboard to work is that the license information about the work must be expressed in RDFa in the source documents.

The design of the Semantic Clipboard is given in Fig 2-4. The tool uses the ‘RDFa Extractor’ to glean the Creative Commons license information expressed in RDFa from the HTML page the user browses. The *UI Enhancer* implements several menu options in the Firefox browser to select licensed images with the proper intention. The available options are given in Fig 2-4. For example, if a user want to see images

that can be used for ‘Commercial Purposes’, she can select the corresponding menu item. Then the images that do not have the CC-NC clause (Creative Commons Non Commercial use) will be highlighted with an overlay on the image. The *Attribution HTML Constructor* is called when the user issues a copy instruction on a particular image by right-clicking on the image and selecting the context menu option *Copy Image with License* as shown in the right half of Fig 2-4. Based on the license information for that particular image, the attribution HTML snippet is constructed as specified by Creative Commons, and copied to the system clipboard. Currently two data flavors are supported: ASCII text and HTML. Therefore if the target application accepts HTML such as in a rich text editor, the source text (with the angle brackets) will not be displayed. The primary goal of this tool is to let users reuse content with minimal effort.

2.6 License Server for Geo Spatial Data⁷

Taking a slight departure from adding support for media (image) reuse, I implemented a way for mashing up two or more raw data sources, specifically map data sources, in an accountable manner.

2.6.1 Introduction

Programmable Web lists 1254 documented mash-up APIs with 6852 mash-ups out of which a highly disproportionate 2281 or 33% are mapping mash-ups, with Google Maps API being the source of 1658 mash-ups itself. Therefore, there is no doubt that mash-ups are very popular, and within mashups, map mash-ups are by far the most popular. The idea of a commons of information has been received with much enthusiasm. Not surprisingly, commons has been proposed as a model for geospatial data as well [62], but it depends on a central repository, and creates many new and unique legal considerations [63]. One proposed solution is to have all the players

⁷The description for this tool is included in the publication titled ‘Policy aware geospatial data’, and was published at the ACM Geospatial Information Systems conference in 2011 [61].

contractually agree to freely open up their data [64].

The Open Geospatial Consortium (OGC) describes Web Mapping Service (WMS) and Web Feature Service (WFS) that allow querying map data over HTTP, thus enabling map applications that draw upon distributed data sets along with local data [65]. In this work I extended the WMS with licensing information.

Since the ccREL-based license is human-readable, and also points to a location on the web from where more information on the license can be retrieved in a plain text file, it can also be stored as a record in a table if the geospatial data are stored in some data source. This makes the license suitable for two of the commonest geospatial data formats. In a typical application, the user serves the dataset via a map server that works in conjunction with a web server to respond to user requests for data. The map server typically gets its guidance from a configuration file that describes all the data layers in the application, their source, selection parameters, and even their styles including symbology, font, colors and other cartographic elements. The map server queries the data sources, extracts the query results, and typically constructs a map image in a standard image format that is then served by the web server.

2.6.2 Design and Implementation

Additions to the Shape file Format

One of the common formats for geospatial data is the Shape file format [66]. The Shape file format stores a single feature type in a collection of files, each one of them holding some aspect of the feature information. The main files for a line feature dataset of roads might be for example:

1. *roads.shp*: the geometry of the lines
2. *roads.dbf*: a table of attributes, one row per feature
3. *roads.shx*: a cross-reference between the shp and the dbf files
4. *roads.prj*: the metadata for the projection information (optional)

5. *roads.qix*: the quadtree index for the dataset (optional)

In addition to the above mentioned files, the Shape file format may have various spatial and attribute indexes and other files. An additional file called *roads.lic* was introduced to this mix, that would hold the license metadata for the dataset. This license would be stored using ccREL which allows expressing a licensing utilizing RDFa in HTML. The user can simply go to the CC license chooser, choose a license through its step-by-step set of screens, and copy and save the license information expressed in HTML as plain text in a *roads.lic* file.

Extending the WMS with a License Server

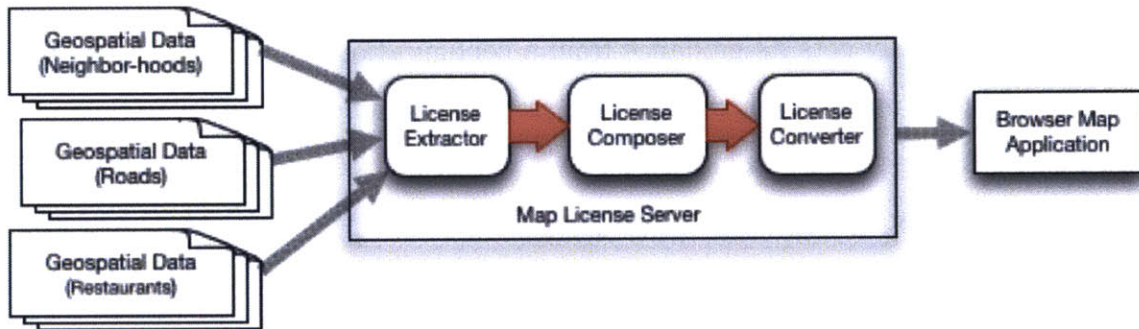


Figure 2-5: Components of the Geo-spatial Licensing Server

We demonstrate a license server analogous to the WMS as shown in Figure 2-5. The license server utilizes a configuration file to query a data set, but instead of querying the geographic features and returning a map image or features, it queries the license information of the data set, processes it, and returns it as an HTML stream. The license server is made up of: a license extractor; a license composer; and a lic2rdfa converter. The user requests the license over HTTP, the license extractor extracts the license from the component datasets and the license composer computes the new license using an efficient license lookup table, such as the one given in Appendix B. Then the lic2rdfa converter creates an HTML fragment that expresses the license using ccREL, and the server sends that information back to the users browser in a machine readable format such as RDFa where that information can be displayed via a link or in a popup window.

An example license lookup table is shown in Table B.1 in Appendix B. The rows and columns of the table constitute the component licenses, and the corresponding cell gives the resultant license if the two licenses can be combined. When we have more than one data source, we can use recursively query the lookup table and construct the resultant license thereby scaling up to multiple data sources.

2.7 Summary

In this chapter, I described a license violations experiment that indicate that there is a strong lack of awareness of licensing terms among content users, and a survey of remix culture on the Web that suggested there is inadequate support in popular websites to help users be accountable when reusing content. To increase awareness of licensing terms associated, I then developed a set of tools that provide policy awareness to the user.

Though these tools help with awareness, they do not provide a complete solution to privacy and copyright protection on the Web because provenance of content as they are repurposed is not preserved anywhere. There is a strong need for an end-to-end accountability architecture on the Web because accountability is the right solution to privacy and copyright protection. This accountability infrastructure is explained in detail in the next chapter.

Chapter 3

An Accountable Architecture for the Web

This chapter describes an end-to-end architecture to enable accountability in web based systems that allows enabling appropriate use and determining inappropriate uses of information to achieve accountability on the Web. The client-side tools described in the previous chapter addressed only one aspect of the accountability requirement, where they enabled appropriate use of information with validation, displaying appropriate use notices, enabling copying of license information with the proper attribution text, and mashing-up two datasets with different licenses. However, it was not possible to determine what happens afterwards to the content, or the source of any usage restrictions violations using any of these tools. Therefore, to enable comprehensive accountability for the Web, I introduce a novel technology called HTTPPA and the supplementary components of it in this chapter.

3.1 Motivation

It has been shown that the ability to copy, collect, aggregate information in large-scale systems and the ease with which it is possible to infer sensitive information in them using publicly available data often results in adverse consequences for users especially in user privacy [67, 68]. Similarly, in intellectual property rights domain

creative content creators have often struggled between making their content available for reuse and remixing while ensuring appropriate use. Even though access control systems alone are often successful in managing access to resources, they are ineffective in preventing information leakages, and provides very little help to users in tracking their information flow across systems. With personal and creative content, users have an incentive to keep track of their content while not restricting access. No one is going to deposit their intellectual output in a repository that will not instill a sufficient amount of confidence in its being around far into the future. Similarly, people are not going to be comfortable giving up their data to an entity that may or may not be around in the future, that may sell or in some other way give away or benefit from their data, for other reasons such as privacy and control. Therefore, we need a technological solution that provides some amount of control in determining how one's information was used, while enabling free flow of information. In the next section, I describe how this vision was achieved using a novel accountable infrastructure for the Web.

3.2 Augmenting the Web with Accountability¹

The Web has become worlds largest distributed application where the contents of databases are increasingly exposed directly. The semantic web facilitates the annotations of these data sets with RDF metadata, forming a global web of Linked Data. The principles of exposing information on the web are now well understood namely the use of Uniform Resource Identifiers (URIs), a system for identifying resources globally, and protocols such as HTTP to access resources. The REST architectural style on the Web provides an architecture that utilizes HTTP and URIs [70]. These architectural constraints emphasizes component interactions, generality of interfaces, independent deployment, and intermediate components to encapsulate legacy systems, and enforce security of web applications.

¹The work described in this section was published in a paper titled 'Augmenting the Web with Accountability at the PhD Symposium, World Wide Web Conference 2012' [69].

However a social constraint such as Accountability needs an **eco-system** that makes it easy for senders to become accountable and the receivers to demand accountability [71]. An accountable protocol on the Web built on the RESTful architecture along with supporting services can provide this eco-system to help alleviate some of the complex problems that arise from information reuse with respect to access, transfer and reuse. This protocol for accountability provides the necessary functionalities to determine appropriate use of data.

Accountable Systems (i.e. servers and clients that adhere to the accountable protocol) should have the following minimal characteristics:

1. Agents, i.e. both users and bots, have a unique identity mechanism.
2. Sensitive data items have usage restrictions associated with them as defined by the data subject or the data provider.
3. Systems interoperate with respect to data, actions, and agents.
4. Usage logs are tamper-evident, stored separately from the data, and provide non-repudiable evidence of the actions by the agents.
5. Users can efficiently retrieve the logs, check them and take action if there are any data misuses.

With these requirements in mind, I designed an end-to-end Accountable Systems architecture as shown in Figure 3-1.

3.2.1 Components of HTTPPA

HTTPPA requires the following four components to work together:

1. **Accountable Client:** A user-agent such as a browser that is capable of understanding usage restriction options sent by the Accountable Server, and responds accordingly with the data transfer.
2. **Accountable Server:** A server side implementation for a specific application, such as a Social Networking website, Electronic Healthcare Records System, or

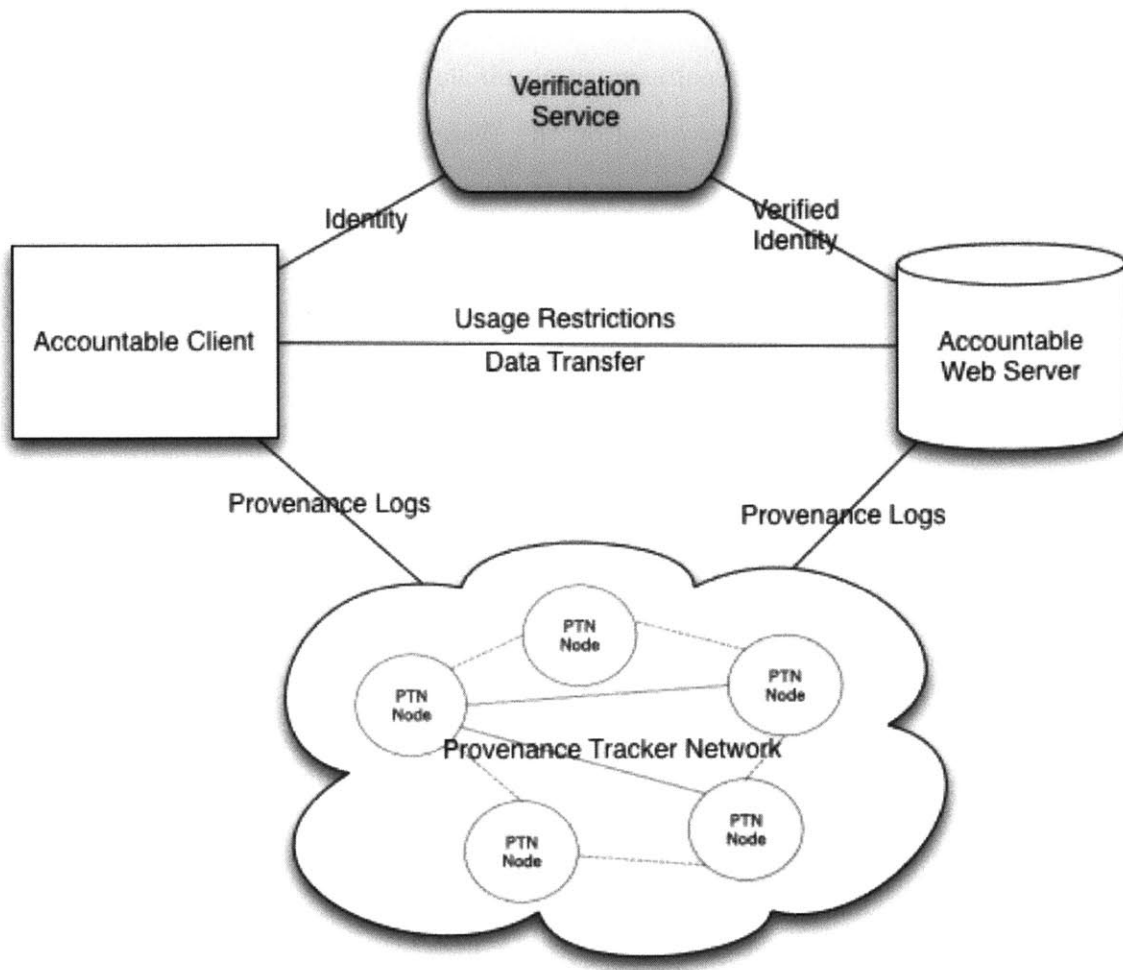


Figure 3-1: Building Blocks of Accountable Systems.

a Photo Sharing Website that understands HTTPPA. The server side application should communicate usage restrictions of the resources it has to the client, and create records of the accesses in the Provenance Tracking Network.

3. **Provenance Tracking Network (PTN):** A network of servers organized as a distributed hash table that provides the means for storing log records. The PTN does not store the actual data items, but rather the pointers of the data items with the data related to the access, repurpose and share activities.
4. **Verification Service:** Provides a unified identity mechanism for authenticating clients and servers to each other and to the Provenance Tracking Network. HTTPPA currently does not enforce a single authentication mechanism, but as

long as there is consensus among the server and the client as to the identity of each party, such an authentication scheme may be used.

3.2.2 HTTPA Mechanism

The data transfer between the client and the server and the usage of it is handled by the HTTPA protocol. HTTPA conveys usage restrictions between web servers and clients, creates a log every time a resource access happens via any of the HTTP verbs, i.e. GET, POST, PUT, etc, and these audit logs can later be retrieved for compliance checks afterwards. An HTTPA transaction will commence when the data provider on the server-side presents a set of usage restriction choices to the data consumer on the client-side to agree upon. This agreement is logged in the PTN. These logs can be later queried by the data subject, after proper authentication, to determine if a breach of usage restrictions has occurred. Similarly, a consumer of content will be able to ascertain the provenance of information using the PTN. Each of these activities will be described in detail in the following sections.

3.3 Specifying Appropriate Use Through Usage Restrictions²

Usage restrictions can be defined as extensions of access control. These take the form of actions that can and have to be performed over data after access has been granted. In HTTPA, the definition of usage restrictions is up to the developers as long as they use a machine readable format such as RDF to describe the terms. I will first describe a motivating scenario before delving into the implementation.

²Some of the work described in this section was published in 'Addressing Data Reuse Issues at the Protocol Level' published at the IEEE International Symposium on Policies for Distributed Systems and Networks [72].

3.3.1 Scenario Walkthrough

In this scenario we take a policy-centric view on social web privacy, where policies capture permissions such as access control, obligations such as terms-of-use and licensing, and other data-handling settings that allow a user to control their interactions with other users. In particular, policies apply privacy settings to the profile and social media frameworks to consistently manage the user expectations of privacy and other obligations. This allows individuals and businesses on the social web to share information without any fear of violating user privacy.

Assume that Alice is a user of an imaginary social networking site called 'SocialBook'. Alice communicates with SocialBook using HTTP, and both parties have specified their intentions and usage restrictions using the linked data vocabulary 'Respect My Privacy' (RMP) described in [73]. The RMP vocabulary describes restriction terms such as 'no-commercial', 'no-depiction', 'no-employment', 'no-financial', and 'no-medical' that conveys a prohibition on exposing the private information for commercial purposes, associated with any pictures, employment purposes, financial purposes, and for treatment purposes, respectively.

Usage Restriction Management on Upload

Suppose Alice wants to upload some pictures on SocialBook. The default settings on her client is set with the usage restriction such that any HTTP payload carrying her data to only be posted if the recipient acknowledges the full ownership of the content to her. However, it seems that SocialBook has extremely draconian terms of service that if uploaded to SocialBook, the data becomes the property of SocialBook. Alice's client notices the incompatibility in these two policies, and informs Alice about it. Alice can then choose to either stop posting her pictures or to request SocialBook her intention of using her own terms of use. In the latter case, Alice will not be able to post the pictures right away to SocialBook. But with some additions to the current HTTP, when someone reviews the request and agrees to Alice's request at a later point, Alice's pictures will be automatically uploaded by the client.

Usage Restriction Management on Download

Alice has a photo on SocialBook with a usage restriction specifying that the photo cannot be used for any *commercial* purposes. An employee from a large advertising company, Bob, downloaded that photo. Bob's client confirmed with SocialBook he agrees to the *non-commercial* usage imposed on the photo.

At a later point, Alice can request the PTN to give a detailed record of any subsequent usages of her photo. Through that she may find out that Bob had used her photo in an online advertisement, or something that may signal a commercial-use. Through her client Alice and sends a request for clarification or a takedown request to Bob with a proof detailing the violation.

3.3.2 Usage Restriction Language

Websites publish privacy policies that communicate planned data handling practices, such as rights of the data, intended purposes of collection, and third parties who may have access to the data collected from the users. Users also have complimentary usage restrictions for what their data can and cannot be used for. We identified several categories of usage restrictions as applied to accountability that are modeled from the needs of individuals rather than from the needs of data silos. These categories are listed below:

- **Temporal Constraints** such as 'delete after 30 days', 'do not publish before 8am on Monday', 'do not show before explicit consent given'.
- **Spatial Constraints** such as 'do not share with somebody outside the organization', 'must not be stored on a server outside the organization'
- **Cardinality Constraints** such as 'can only embed 1 time', 'can only be downloaded to 5 devices', 'can only share with 100 people'
- **Fulfillment Conditions** such as 'attribute upon usage', 'takedown upon notice', 'do not share upon notification', 'notify if a reuse of this work is shared'

- **Dissemination Constraints** such as ‘public’, ‘within organization’, ‘private’
- **Purpose** such as ‘commercial’, ‘educational’, ‘research’

The RDF vocabulary describing this language with the categories mentioned above is available at <https://raw.githubusercontent.com/mit-dig/httpa/master/vocab/ur.ttl> for reference. One of the challenges in defining such a usage restriction language was that many of these categories are domain dependent. A side effect of the use cases and reference applications that are focussed on privacy and intellectual property rights protection was that the vocabulary came out to lean heavily towards these domains. However, as with any linked data vocabulary, this vocabulary is open for extension by the community.

3.4 Usage Restrictions Management

Usage restrictions need to have three important characteristics: they have to be easy to produce, be co-located with the data they describe, and be easily readable. The easiest way to achieve this is to have them be produced automatically. Any usage restriction that has to be produced manually by the user usually doesn’t get produced at all. The easiest way to ensure that the link between the usage restriction metadata and the data they describe is not broken is by embedding the former inside the latter, or sending them together when the data or the resource is requested. This way, the two travel together. Finally, usage restrictions have to be accessible easily, readable both manually as well as programmatically.

Managing usage restrictions in HTTPPA includes both communicating them from the information provider to the information consumer, as well as validating them to check if they match with the intentions of data access, use or modification.

To communicate the usage restrictions, the provider sends the usage restriction options available with the resource to the consumer. These usage restriction terms are sent in an HTTP header called “Usage-Restrictions”. Depending on the policy set on the data consumer, it may convey the acceptance of the usage restrictions to

the data provider in an “Usage-Restrictions-Agreement” message, or request for a change in the usage restrictions.

3.4.1 Requesting a Resource with HTTP GET

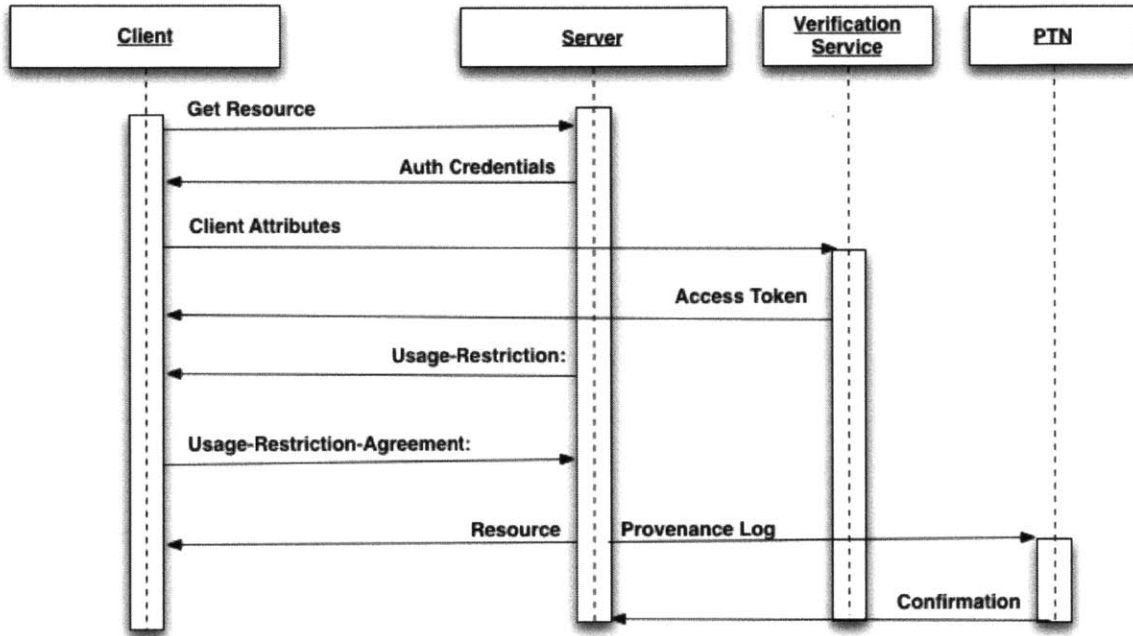


Figure 3-2: Sequence Diagram Resource Access in HTTPPA.

Handling usage restrictions for an HTTP GET assumes that there is an existing resource. If there is no such resource, the usual HTTP semantics comes into play, i.e. the response being ‘Resource Not Found’ with the 404 code. If as expected, the resource exists, the server intercepts the usual HTTP response pipeline, and will send the usage restrictions associated with the resource after authenticating the user with any global identity mechanisms such as WebID [74] or OAuth [75]. The client send the ‘Usage-Restriction-Agreement’ header along with the resource that it originally requested, and the usage restriction. Upon receiving this response, the server verifies that the usage restriction sent in the response matches the stated usage restriction for the resource on the server. If it matches it will send the resource to the client, and generates a log record on the PTN.

3.4.2 Creating a Resource with HTTP PUT and POST

In the case of HTTP PUT and POST transactions, the server becomes the data consumer and the client becomes the data provider. Similar to the GET scenario, the client will need to be authenticated with the server. Both these methods request that the enclosed entity be stored under the supplied Request-URI. If the Request-URI refers to an already existing resource, the enclosed entity should be considered as a modified version of the one residing on the server. If the Request-URI does not point to an existing resource, and that URI is capable of being defined as a new resource by the requesting user agent, the origin server can create the resource with that URI. The server will also store the usage restrictions associated with this resource to be communicated on subsequent accesses to this resource. If a new resource is created this way, the origin server must inform the user agent via the 201 (Created) response. If an existing resource is modified, either the 200 (OK) or 204 (No Content) response codes should be sent to indicate successful completion of the request. If there is an existing resource, there is a possibility for a usage restriction mismatch. In such a case, the server will send an additional header indicating 'Usage Restriction Mismatch' to the client. The client may decide to discard the header in which case, the server will use the last value for usage restrictions sent by the client.

3.5 Provenance Tracking and Auditing

An important research question that stems from the earlier work outlined in the previous chapter is the need for a method of provenance preservation as data is being shared and reused. Therefore, in the context of our work, tracking provenance is perceived as a critical issue, and this section describes how we have implemented it.

3.5.1 The Provenance Tracking Network (PTN)

Core of our provenance tracking and auditing infrastructure is the 'Provenance Tracking Network', which we describe in detail below.

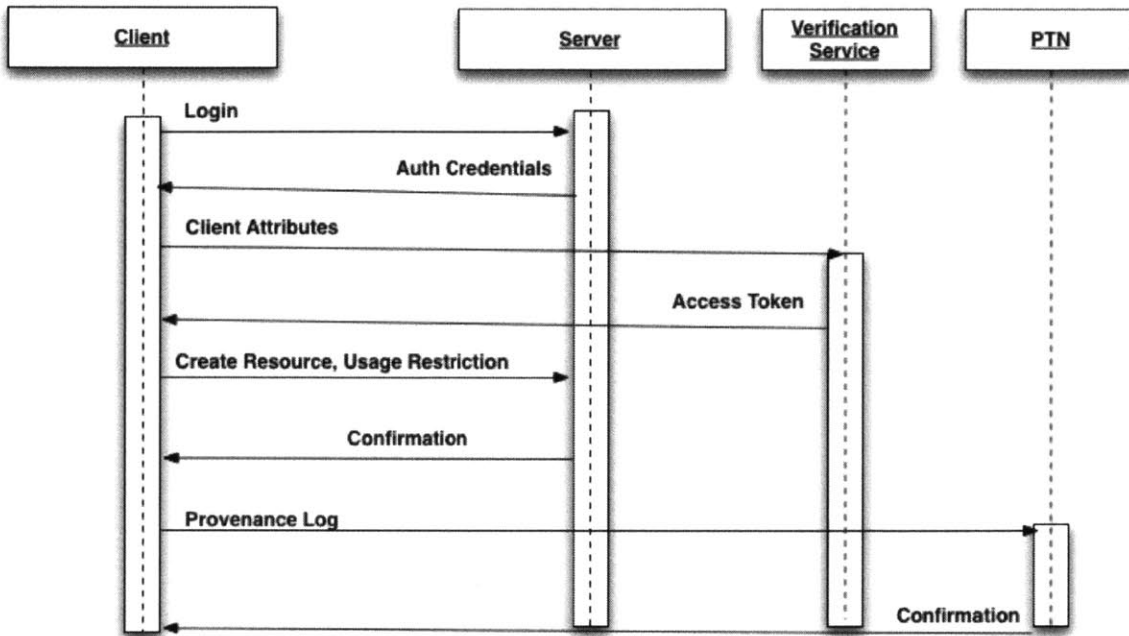


Figure 3-3: Sequence Diagram for Non-Existing Resources in HTTPA.

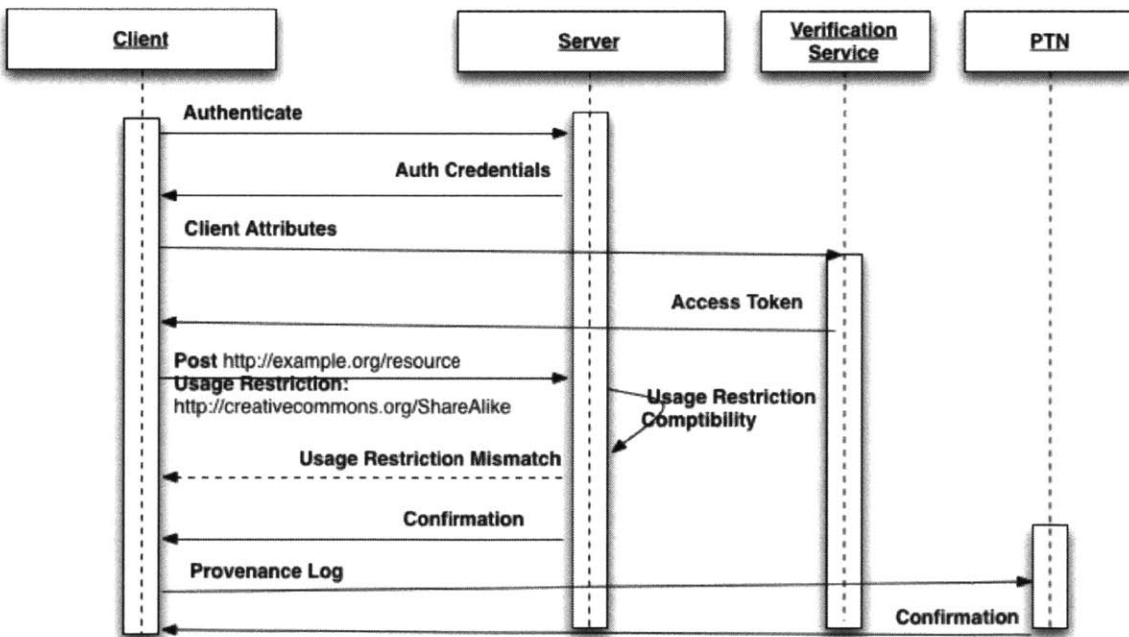


Figure 3-4: Sequence Diagram for Existing Resources in HTTPA.

Design Considerations for the PTN

Provenance as a key enabler for accountable systems since it consists of an explicit representation of past processes, which allows us to trace the origin of data, actions

and decisions (whether automated or human-driven). Provenance also helps determine the quality of trust one can put into the data. This can be provided through necessary logs that can be accessed very efficiently to reason about compliance or violations. Another important consideration in designing the provenance infrastructure was not to rely on just one single authority to store the provenance logs, but rather to implement it on top of a collection of peer servers that can join and leave the system any time they desire, thus democratizing the responsibility of maintaining accountability.

Overview of the PTN

The PTN is implemented as a decentralized network of peer servers that maintain the access and usage logs for data of interest, i.e. sensitive data or creative content. No single entity can exercise ownership over the entire collection of log records. Only the owner or the data subject of the sensitive data item included in the log record will be able access it from the PTN. Since it is implemented using a distributed hash table (DHT), by design the system is fault tolerant, and the lookups are fairly efficient. DHTs support the simple put/get interface from traditional hash tables, but also offer increased capacity and availability by partitioning the key space across a set of participating nodes in a network. By enabling the ownership of log records to be held by a collection of peer nodes rather than a single centralized server helps ensure that the provenance logs cannot be tampered with. Checksums of the log records from different peers are compared periodically and the peers that are known to tamper the records rejected for integrity. A centralized design may very well work for reliable storage of usage data and to generate audit logs to enable privacy in a transparent manner. However, a decentralized design is better due to couple of reasons: (1) Any attempt to tamper the log records will result in a modified checksum, and the node is can be easily identified when the records are updated, and that particular node can be penalized. If the data is stored centrally, the log records can be changed at the hosting servers' discretion. (2) A decentralized architecture of the PTN replicate the data at many nodes, thus a failure of a single node would not affect the overall

performance.

Servers that wish to participate in HTTPPA can join via the published gateway nodes. Some of these PTN gateway nodes are listed at <http://csail.mit.edu>.

Implementing the PTN

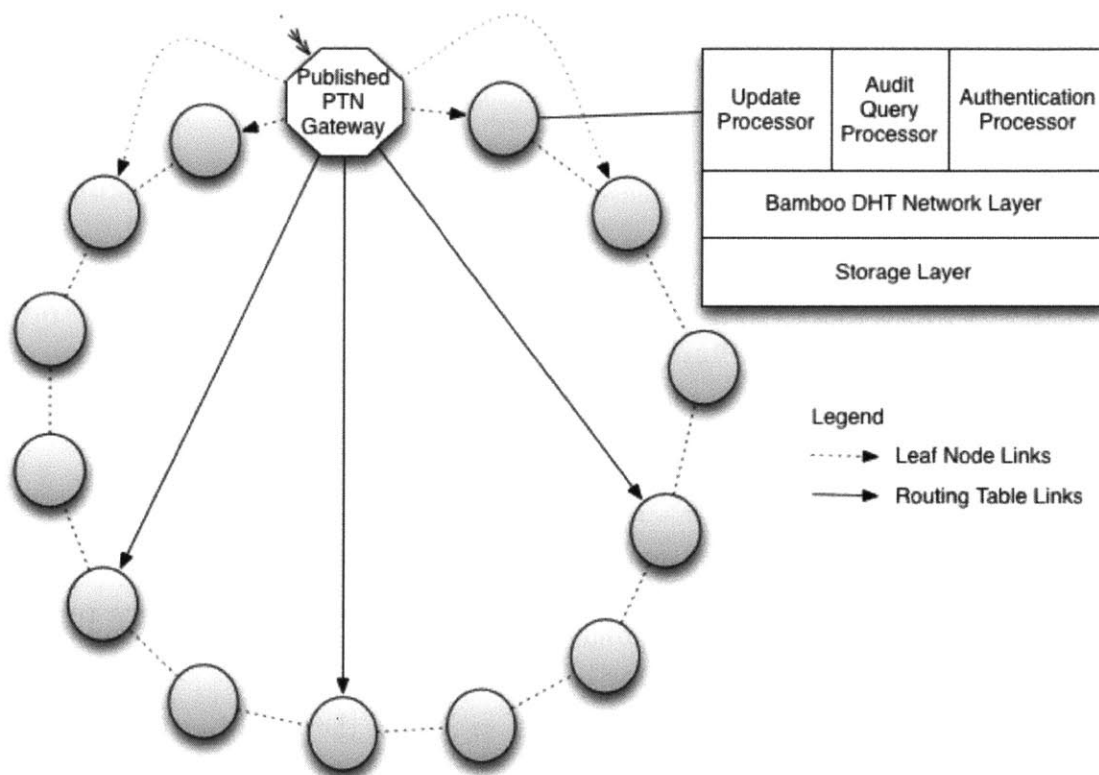


Figure 3-5: The Provenance Tracking Network architecture

The PTN implementation in HTTPPA was motivated by OpenDHT [76]. OpenDHT is fairly limited in its programming interface, as it only has **unauthenticated** support for `get` and `put` methods. Also, since it was designed to be a public DHT service that can be used by untrusting services and clients, OpenDHT's storage mechanism does not persist the records. For the PTN implementation, the DHT overlay used in OpenDHT was extended and added the following: (1) The ability to add, update, and retrieve authenticated records with the help of the Authentication, Update and Audit Processors and (2) A persistent storage mechanism using the Log Store. The PTN only stores the log records of the data, and not the actual data items.

As shown in Figure 3-5, each node in the PTN has a persistent storage for the log records. the Bamboo DHT Overlay Layer [77] was used to find the routing table links to connect to nodes that may not be adjacent in the DHT key-space. The Update Processor appends to provenance log records and propagates the log to other PTN nodes that may have the same entry. The Audit Query Processor performs the retrieval operations in the PTN, while the Authentication Processor ensures that the requestor for the provenance logs is either the subject or the owner of the data item the provenance log is created for.

3.5.2 Accountability Logs

There are complex data interactions at play in many web information systems. These data interactions include the agents and processes that consume the data, as well as the usage restrictions and intentions imposed on the data. In modeling the data, a potential sensitive data item can be marked as such by attaching a usage restriction to it. Every time an agent in the system accesses, updates and transfers the sensitive data item through a process, a log record is created in the PTN by the Accountable Web Server.

Accountability Logs have several characteristics: they are immutable except by protocol components, secure, readable only by trusted parties involved in the HTTP transaction, and have all the records pertaining to a particular data transfer and usage, such as: what data was accessed, the specified intent of access, and the agreed upon usage restrictions.

The log record contain the triple consisting of: a key (k), a value (v), and the hash of a chosen secret up to 40 bytes in length (H). k should be a 160-bit value and the v can be variable-length and there is currently no restriction on the size. All the usage log entries to the PTN $\langle k, v \rangle$ are persisted in a datastore at each peer node in the PTN.

Log Composition

For describing the value v of the entry in the PTN, i.e. the accountability log record, the W3C Provenance Ontology (Prov-O) recommendation [78] was used. The provenance graph consists of RDF triples consisting of the following terms defined in the Provenance Ontology.

- *prov:Activity* - Any action in the HTTPPA server or the client that results in accessing, repurposing, or sharing of information that has a usage restriction attached with it.
- *prov:Entity* - The piece of information that has a usage restriction attached with it.
- *prov:Agent* - A physical thing, i.e. a human, or an organization, that carried out an activity.
- *prov:atTime* - The time at which an activity occurs in HTTPPA.
- *prov:influencedBy* - The relationship between two activities that results in an information exchange.
- *prov:generated* - The relationship between an activity and entity that results in the latter being generated as a result of the former.
- *prov:wasDerivedFrom* - The relationship between two entities, where the source is identified.

Creating Usage Logs

First, the PTN attempts a 'get' on the same key. If there is an existing key in the PTN, those values are retrieved. As in any DHT implementation, our PTN is also susceptible to churn. Therefore, some PTN peers might have an older version of the provenance log for a given data item because it went offline when a previous update was received. Therefore, our algorithm checks the values of all the keys retrieved

within an allotted time. This time can be configured by the application developer, and defaults to 5 seconds. We use the values of the `prov:atTime`, as defined in the provenance ontology [78], in the RDF graphs retrieved for the given key to determine the newest entry. Once such a value was determined, the new triples are appended. This new `(key, value)` pair is then propagated in the PTN, and the peers that receive this `(key,value)` pair will either add it as a new entry or replace a previous entry by the same key.

PTN Deployment

In order to gauge the effectiveness of this infrastructure, Provenance Trackers were deployed as an overlay network on PlanetLab, an open platform for developing, deploying and accessing planetary-scale disruptive services on the Internet [79]. Disregarding the disruptions that are often associated with a free research oriented network, we found that this PTN deployment, has a low churn rate, and the nodes have near perfect uptime. We have also set up a gateway node that always connects to the PTN deployed at MIT CSAIL at <http://csail.mit.edu>, and expect that volunteers will run the provenance trackers and enable the growth of this network.

Using the PTN

Any web application that needs to interface with the PTN can use the PTN wrapper interface shown in Figure 3-6. We introduce the concepts ‘Agents’, ‘Sensitive Data’ and ‘Processes’ within the wrapper interface that can be configured by the application developer. We have contributed two reference implementations in the Django python web framework and in node.js as middleware modules³.

3.5.3 Accountability Checking

A browser button called ‘Oh Yeah?’ which is used to provide reasons why the user should trust the data was envisioned by the founding father of the Web in 1997 [80].

³The source code for these libraries are available at <https://github.com/mit-dig/httpa>.

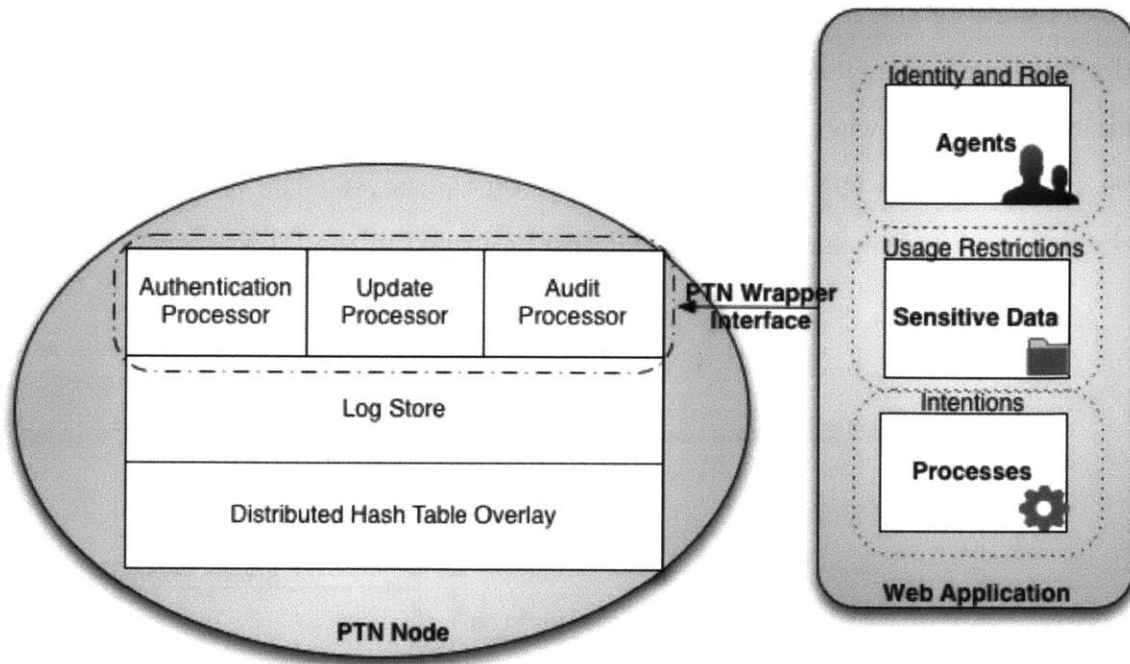


Figure 3-6: PTN and an Accountable Systems application interact through the PTN wrapper interface.

We envision such a button on Accountable Systems, where users can ask ‘Oh Who?, Oh What?, Oh Why?, Oh When? and, Oh Where?’. We implemented a similar functionality with the ‘Audit’ Button in Accountable Systems that triggers queries on the PTN. In short, if a user finds that her data was misused and / or the usage restrictions associated with it were violated, she can take recourse by producing a provenance trail with the help of the PTN. It first verifies that the requestor of this information is indeed the owner of the resource. Then it performs several DHT lookups to create a trail of transactions involving this resource. An example of this button is discussed in detail in our use case on Transparent Health in Chapter 4.

The sequence of actions that takes place when the ‘Audit’ button is clicked is represented in Figure 3-7. First the data owner, i.e. the agent that can prove ownership to the sensitive data at the data provider, makes an audit request. If the data owner is not already authenticated with the data provider she will be redirected to be authenticated via the Verification Agent. Once authenticated, the data provider issues a get request to the PTN. The authentication processor in the PTN will validate the

authenticity of the request, and if validated, will send the provenance log record for the sensitive information that was requested. Since the identity of data consumer is known, the data owner can request for clarifications. The data consumer can either be another user in the system or a process on the server. The process of clarifying the data access, use and transfers will also be logged in the PTN.

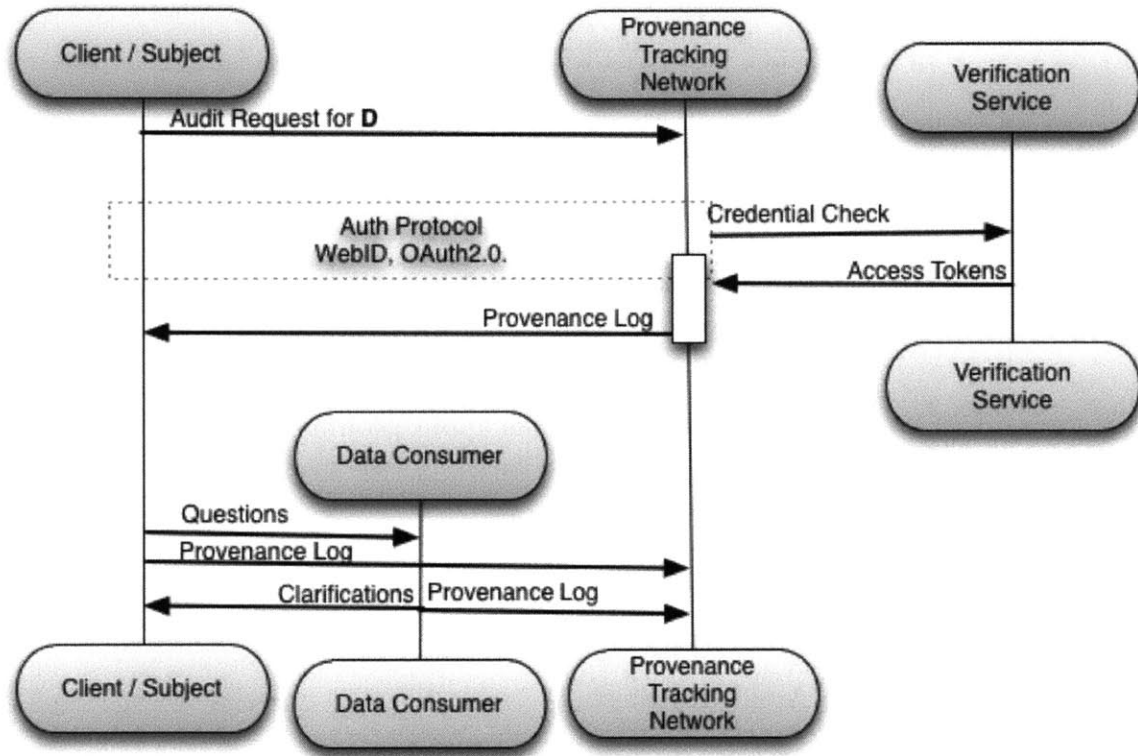


Figure 3-7: Auditing Usage Logs with the PTN

Scenario Walkthrough

To illustrate accountability checking with HTTPPA, let us consider a modification of the scenario described in Section 3.3.1. Suppose Alice uploaded a photo on a public photo sharing website with a usage restriction specifying that the photo could not be used for any *commercial* purposes. Bob, the employee from a large advertising company, accessed that photo with the intention of using it for personal use. Bob's client confirmed with the server that the intention of accessing the photo was *non-commercial*, and that he would honor Alice's usage restriction. Bob modified Alice's

photo slightly, and reposted it on his personal website along with the usage restrictions for *non-commercial* use set by Alice. Carol, another employee of Bob's advertising company sees this picture, and uses it in an advertisement for the company. When accessing the photo, Carol uses an HTTPPA-aware client, but when modifying the photo in order to use it in the company advertisement, ignore the usage restriction attached with it by mistakes.

Few weeks later, Alice found out that her photo was used in an online advertisement, and she is interested in knowing how her usage restrictions were violated. With HTTPPA, Alice can request the provenance tracker network to construct an 'audit trail' for her by giving the URI of her photo on the photo sharing site that has been inappropriately used for a commercial purpose. The PTN verifies that Alice owns the resource and looks up accountability logs within the provenance tracker network. It verifies that Bob had agreed to the original usage restriction that Alice had set, he had made some modifications to the photo, reposted elsewhere with Alice's original usage restrictions, and then Carol had accessed this image from Bob agreeing to honor the usage restrictions, but had failed to do so. Alice can now request the provenance trackers to send Carol a proof detailing the violations, and using the identity information available in the audit record ask for a takedown, since the advertisement violated her terms of use of her original photo.

As was mentioned before, HTTPPA and the corresponding Accountable Systems that are built with it, assumes the end-to-end arguments and non-enforcement. Therefore, it is allowed for Carol to use a HTTPPA-aware client to access the resource, but use a non-HTTPPA-aware client to modify and share/upload it. In this case, the provenance trail breaks, and it would be difficult to pinpoint the individual that may have violated the usage restriction.

3.5.4 Performance Evaluation

We envision the PTNs to be deployed at a global scale serving many web applications. To achieve this goal, we want the PTN to be robust, fast, and handle huge loads while supporting its growth organically. We conducted an initial experiment using

PlanetLab [79], to gauge the performance characteristics of the PTN. The primary goal of the test was to measure the average time it takes to complete a ‘get’ operation of an authenticated key that identifies a data record, and similarly the average time it takes to perform an authenticated ‘put’ operation to insert a given provenance log record in the PTN.

Measurement Setup

We selected 100 nodes and deployed our PTN code to simulate a large scale PTN deployment. Within a 24 hour period, we continuously assessed the latency of gets and puts based on a synthetic workload from a script that issued a get and a put every second. For a get request the *time* measurement was the time elapsed between the request and the retrieval of the value from the PTN. Similarly, for a put request the *time* measurement was the time elapsed between the request and the acknowledgement *success* from the PTN. For the put operations, the synthetic workload included several different kinds of provenance logs, (1) new provenance records to simulate the creation of a new entry in the PTN, (2) existing provenance records that includes one value of `prov:atTime` in the RDF graph, and (3) existing provenance records that includes multiple values of `prov:atTime` in the RDF graph

Results and Analysis

Since we issued put and get requests every second for 24 hours, we obtained 86400 data points for each type of request. To simplify the analysis, we averaged the response times received within an hour for each request and plotted the values as shown in Figure 3-8. The results indicate that the put requests are more time consuming compared to the get requests, unlike in traditional DHT environments such as OpenDHT, where the time to complete the operation is approximately similar for both types of requests [76].

We also analyzed the performance of the PTN to determine the characteristics of the architecture when applied in a large scale deployment.

Requests (both put and get) in the PTN are authenticated. Therefore, a little

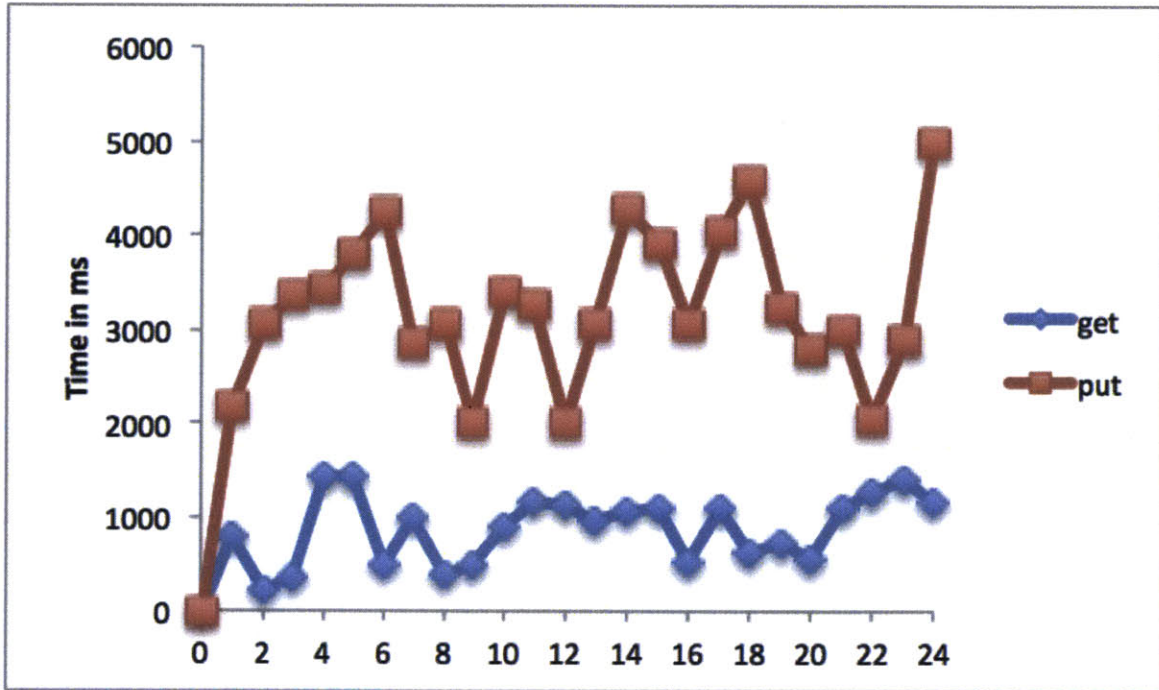


Figure 3-8: Average time it takes to complete authenticated get and put requests in a large deployment of a PTN consisting of 100 nodes during a time span of 24 hours

bit of extra time is spent on authenticating the provenance log requesting service. However the disparate difference in timing between the get and the put requests is due to the fact that, for the put request, our algorithm first performs a get on the given key to retrieve existing values, subjected to a timeout which was set to 5s in the experiment. Then from the values received, the algorithm picks the most up-to-date record based on the `prov:atTime` values in the graph. We can infer that these extra steps carried out before the put is updated on the PTN, to be the culprits for the extra time. However, we did not notice a significant difference in the time for the three kinds of provenance log records that were generated synthetically, indicating that our algorithm is invariant under different input types.

Another interesting observation was that at the end of the experiment only 78 of the original PTN nodes were functioning. The reasons for the missing nodes were server reboots, and resource limitations. However, we did not notice a significant change in the latency of the get and put requests when these nodes left the PTN. This indicates that given a sufficiently large number of nodes in the PTN, the performance

is not susceptible to churn. We claim this to be a salient feature for an architecture that is designed to be deployed at web scale.

3.6 Verification Service

Every agent must have a unique identifier, so that the agent can be identified within the system, as well as in the PTN to ascertain who accessed, used, transferred data in case of compliance checks after-the-fact. Traditional username and password mechanisms do not suffice in our decentralized architecture, as the agents may be acting in different username/password mechanisms. Therefore, in HTTPPA we use a ‘Verification Service’ to delegate authentication.

We recommend using the Semantic Web based approach for handling global identity using the WebID access control delegation as defined in [74]. A WebID is a URI that refers to an agent, when dereferenced identify the agent that it represents. The WebID protocol enables global identification of agents using asymmetric cryptography. The origin server, the server where the WebID is hosted, controls the identity of the agent. When an agent needs to authenticate himself to the PTN, the Verification agent can be delegated to do the authentication of the user. The browser based provenance enabled client will prove the possession of or access to a private key, whose corresponding public key is tightly bound to the WebID that is being authenticated. The private key is associated with an X.509 certificate on the user’s computer, and the public key is associated with the agent’s WebID profile.

In our user studies and in the implementations described in Chapters 4 and 5, we supported the more mainstream authentication approach with the OAuth 2.0 protocol [75]. The agents in these Accountable Systems used Google OAuth services as the Verification Service.

3.7 Summary

In this chapter, I described an architecture that enables accountability on the Web. This architecture is driven by the HTTP with Accountability (HTTTPA) protocol, and is supplemented by the Usage Restriction descriptions, Provenance Tracking Network (PTN) and Verification Service. Clients and servers can implement HTTTPA for a specific task, and use the PTN infrastructure for logging activities on any resource that has a usage restriction attached. I have conducted a performance evaluation of the PTN that confirms that the infrastructure does not have any bottlenecks and there are no technical barriers for it to scale. In the next two chapters I will discuss reference accountable systems that implement this architecture.

Chapter 4

Use Case 1 - Privacy Enabling Transparent Systems

This chapter describes an proof-of-concept application of HTTPPA in healthcare. We developed a healthcare record sharing and modification system called **Transparent Health**¹ with select features from those systems that are used in real world hospital environments.

4.1 Brief Introduction to EHR Systems

Electronic Health Record (EHR) systems promise a wide variety of benefits and capabilities for healthcare. Through EHR systems health care providers can easily send and receive patient data necessary for treatment and analysis, and the patients themselves can use their data to track their health conditions. But the technologies that make these capabilities possible, brings with them some undesirable drawbacks. Patient privacy concerns resulting from unauthorized uses of sensitive patient data was among one of the highest concerns for the newly enacted EU directive to enable all Europeans to have access to online medical records [82]. Solutions through preventive measures often conflict with information requirements of care providers. Therefore, it

¹The work described in this chapter was published at the IEEE Privacy Security and Trust conference in July 2013 in a paper titled 'Enabling Privacy Through Transparency' [81].

is important to achieve a proper balance between these requirements to make health data accessible to patients.

Furthermore, there is a plethora of free apps for nearly every health problem. Unfortunately, in the US for example, these apps are not covered by the privacy provisions of the Health Insurance Portability and Accountability Act or HIPAA², because they do not fall under the category of 'covered entities', unlike the health information shared directly between the patient and the doctor or the hospital. Also, according to the HIPAA privacy rule, patients have the right to inspect and obtain a copy of their entire medical record with the exception of psychotherapy notes. A patient also has the right to an accounting of disclosures of protected health information made over the past six years [6]. However, the support for providing the data in an electronic medium is not that prevalent [83].

EHR Systems usually integrate data from various sub-systems within the hospital EHR System. Some of these sub-systems include, for example, the pharmacy, radiology department, health insurance office, etc. Additionally, these EHRs interoperate with other hospitals with potentially limited capabilities when it comes to referrals.

Many access control system utilized in hospital environments, exercise optimistic security, because preventing access to information may have undesirable consequences. However, in the wrong hands, these over-broad permissions may result in privacy violations. To circumvent this issue, we have developed Privacy Enabling Transparent Systems (PETS) using HTTPA that makes transparency a key component in systems architectures. PETS enables data consumers to be transparent with regard to data usages and determine if there has been privacy violations after the fact. We conducted a user study on a healthcare information application built using PETS to see if transparency on access and usage data satisfies their intentions about the usages of the data.

²HIPAA privacy rule is available at <http://www.hhs.gov/ocr/privacy/index.html>.

4.2 Scenario Walkthrough

This use case illustrates how providing patient privacy can be addressed through PETS. Our solution also addresses the issue of fragmented data on the Web. For example, a patient's data may be located at her primary care provider, at a specialist's office, the pharmacist, the insurance company, and with some third party apps. Our transparent architecture is inherently decentralized, such that, data that is fragmented across all these entities can be modeled and integrated into a unified view.

The 'agents' in our use case include, say, *Patient Peter*, *Doctor Dee*, *Steven Special*, *Pharmacist Precilla*, and *Insurance-agent Ira*. In addition to these human agents, *Patient Peter* also interacts with the free health app *MyHealth* that has *Peter's* health information such as age, height, weight, blood pressure, cholesterol levels, vaccination data, medical conditions and medications both past and present. These agents operate in different systems. For example Doctor Dee who works at the General Hospital is Peter's primary care provider and Peter's primary health records are kept in a database at the General Hospital. Steven Special works at the Star Hospital, and Peter was recently referred to Steven Special by Doctor Dee. Steven Special had to request all of Peter's medical records from General Hospital to get a comprehensive overview of Peter's conditions. However, the medical records pertaining to the referral visit was stored at a database in the Star Hospital. Pharmacist Precilla, when filling the prescription, always looks at Peter's allergy information and past medications available from Peter's health record from the General Hospital, and now she refers to the records available from Star hospital as well. Insurance agent Ira receives health insurance claims from the General Hospital and Star Hospital for the procedures, as well as laboratory tests performed on Peter, and another claim from the Pharmacist for the medications. Depending on the health insurance policy, Ira may even request Peter's complete health profile to process the claim information. The MyHealth app might aggregate Peter's health information, daily activities, eating habits and sell all that to an online advertising company.

As illustrated by the scenario above, there are complex information flows between

various agents in these systems. There is always room for information misuse even if these agents are authorized to access, use and transfer the information. Therefore, Peter might have a legitimate concern that none of the agents in these systems use the information other than for his treatment purposes. This concern might be aggravated if Peter is a celebrity and tabloids have an interest on his sensitive health information. The same applies if he is employed at the hospital, where his bosses and co-workers who have legitimate access to the system can pry on his private health information. Peter cannot enforce any preventive measures on the data usages as there could be emergency override situations where Peter might not be conscious or available to give meaningful consent for usage of his sensitive health data. However, if all accesses, usages and transfers of the data are recorded and are accessible to Peter, he will have a better trust in the system. The agents who use the sensitive information can use compliance checks after the fact to see if they have not violated any rule, or to see if there has been any mistakes. With such a transparency mechanisms in place, we can ensure that web based information management systems can be privacy preserving without being overly preventive.

4.3 Generating Provenance

In this section we model the interactions described in the previous section in a fictitious EHR to demonstrate decentralized provenance management with the PTN in distributed accounting of sensitive data disclosures.

4.3.1 Modeling the Data

Suppose Doctor Dee, a primary care provider working at the General Hospital, sends Patient Peter's personal medical records to Steven Special, a specialist working at the Star Hospital. The original data record for Peter may be available in a data store at the General Hospital, accessible at <http://genhospital.org/patient/peter/medicalrecord>. Steven will need to refer to that record to fully understand Peter's conditions before treating him. Since Steven Special does not have access to that

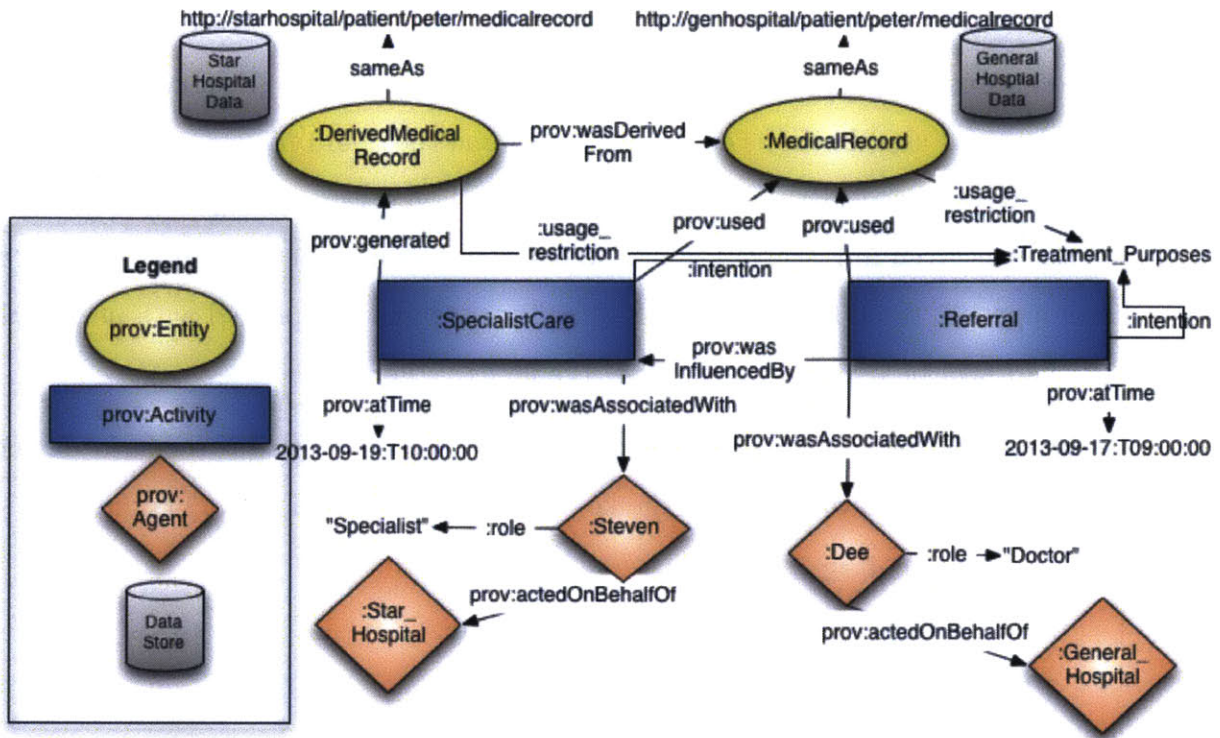


Figure 4-1: Provenance structure of data when a record was sent to another agent acting on another system. In this example, Doctor Dee refers a patient to Steven Special. For treatment purposes Steven needs the entire health record from General Hospital to be viewed.

record, Doctor Dee sends the medical record to him. Steven will use the information from this record, and update the findings from Peter’s visit in the Star Hospital’s data store, where the medical record is available from <http://starhospital/patient/peter/medicalrecord>. The structure of the data including the derived data in this scenario is shown in Figure 4-1. In modeling the scenario outlined above, we used the Provenance Ontology that is now a W3C recommendation [78] and note that everything that is defined in the provenance namespace is qualified with **prov:**.

4.3.2 Modeling the Events

The timestamp of the event is given by **prov:time**, and the events, i.e. **prov:Activities** are associated with a corresponding actor who may in turn be acting on behalf of another agent, i.e. their place of work for example. This graph can be serialized

into RDF triples to represent the state of the Peter’s medical record at the point of the two processes `:referral` and `:specialistCare`. The `:MedicalRecord` data item that was used in the `:referral` process and the `:DerivedMedicalRecord` data item that was used in the `:specialistCare` will generate corresponding entries in the PTN as given in Listings 1 and 2. When the second event, i.e. `:specialistCare` has happened, the PTN entry for `:MedicalRecord` will be updated with new triples, and there will be a new entry for `:DerivedMedicalRecord`.

Listing 4.1: Provenance Generation in the PTN for the ‘*referral*’ scenario depicted in Figure 4-1. `:MedicalRecord` is the key and `<graph>` given in `{...}` is the value

```

:MedicalRecord ->
{ :Referral prov:used :MedicalRecord;
  prov:atTime "2013-09-17:T09:00:00"^^xsd:DateTime;
  prov:wasAssociatedWith :Dee;
  :intention :Treatment_Purpose.
:MedicalRecord owl:sameAs <http://genhospital.org/... >;
  :usage_restriction :Treatment_Purpose.
:Dee :role "Doctor";
  prov:actedOnBehalfOf "General Hospital".
}

```

Listing 4.2: Provenance Generation in the PTN for the ‘*specialistCare*’ scenario depicted in Figure 4-1. Note there are two entries here for both the `:MedicalRecord` and the `:DerivedMedicalRecord`

```

:MedicalRecord ->
{ :DerivedMedicalRecord prov:wasDerivedFrom :MedicalRecord;
  :SpecialistCare prov:used :MedicalRecord;
}

:DerivedMedicalRecord ->
{ :SpecialistCare prov:used :MedicalRecord;
}

```

```

prov:generated :DerivedMedicalRecord;
prov:atTime "2013-09-19:T10:00:00"^^xsd:DateTime;
prov:wasInfluencedBy :Referral;
prov:wasAssociatedWith :Steven;
  :intention :Treatment_Purpose.
:DerivedMedicalRecord owl:sameAs <http://starhospital.org/...>;
  :usage_restriction :Treatment_Purpose.
:Steven :role "Specialist";
prov:actedOnBehalfOf "Star Hospital".
}

```

4.3.3 Securing Logs in the PTN

When the log record described above (k) is ‘put’ in the PTN, it is encrypted using the hospital server’s key (K_S), i.e. $\sigma = \{H(k, v)\}_{K_S}$. A ‘get’ in the PTN should specify both the URI of the sensitive data item this log record is for (k) and the hospital server’s public key ($H(K_P)$). This would only return values that match both k and $H(K_P)$ after authenticating the server using the credentials presented upon accessing the audit log.

The two ‘get’ and ‘put’ operations on the PTN used to model this scenario is summarized in Table 1.

Operation	Returns	Functionality
put(k, v, K_P, σ)	success	Write (k, v), K_P , and $\sigma = \{H(k, v)\}_{K_S}$
get($k, H(K_P)$)	{ v, σ }	Read v stored under ($k, H(K_P)$)

Table 4.1: Secure Operations for Usage Logs on the PTN

4.4 Transparent Health

To evaluate the viability of PETS we designed and implemented an EHR system called *Transparent Health*³. In this system, each access, use, and transfer on a health care record data marked as ‘sensitive’ is logged using the PTN implementation described in Chapter 3. This system also models data from different systems. For instance, the patient’s demographics and medical conditions data are stored at his primary care provider’s information system, the medications information is stored at the pharmacist’s information system, and the referral information is stored at the specialist doctor’s information system. When a user joins Transparent Health, they can pull in data from these different systems to obtain a unified view as shown in Figure 4-2. Next to each ‘sensitive’ information there is an ‘Audit’ button that the user can obtain more information about that data usage. For example, as can be seen in Figure 4-2 Peter Patient’s medical report indicates that he has HIV/AIDS, a sensitive medical condition, and he might be interested in knowing if his medical conditions was only used in connection with his treatment purposes, and not for other purposes such as employment or insurance purposes.

By clicking on the “Audit” button next to the HIV/AIDS medical condition, Peter can check to see how that has been consumed by the various agents in the system. Figure 4-3 shows an example audit log that will be displayed to Peter. It gives the date the event happened, who is responsible for the event, the role of that agent, and the purpose/intention as stated by the agent. If the patient thinks that this is a suspicious/unwanted access of their data, they can request for clarification from the agent by submitting a “Question”.

4.5 User Study on Transparent Health

One of the primary enablers of PETS is the ability to determine when, where, how, what, why an individual’s privacy was violated, and who is responsible for the violation. Therefore, by using Transparent Health as a test bed, we conducted a qualita-

³Transparent Health is available at <http://www.transparent-health.us> for demonstration purposes.

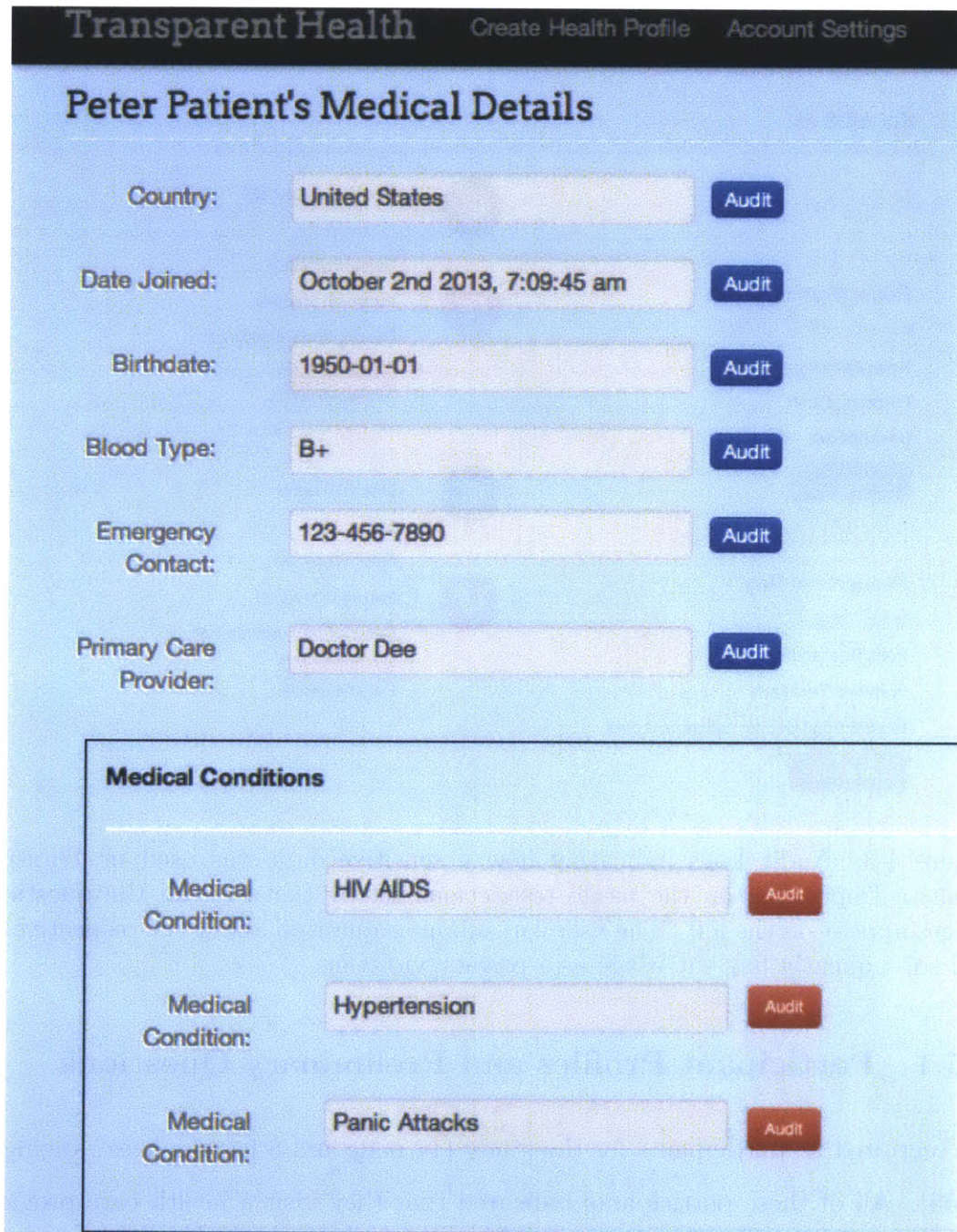


Figure 4-2: Transparent Health: The complete health record of patients with the red 'Audit' button next to potentially sensitive information.

tive study on user perceptions about having access to usages of their data in such a transparent manner.

Audit Medical Conditions

HIV AIDS 4

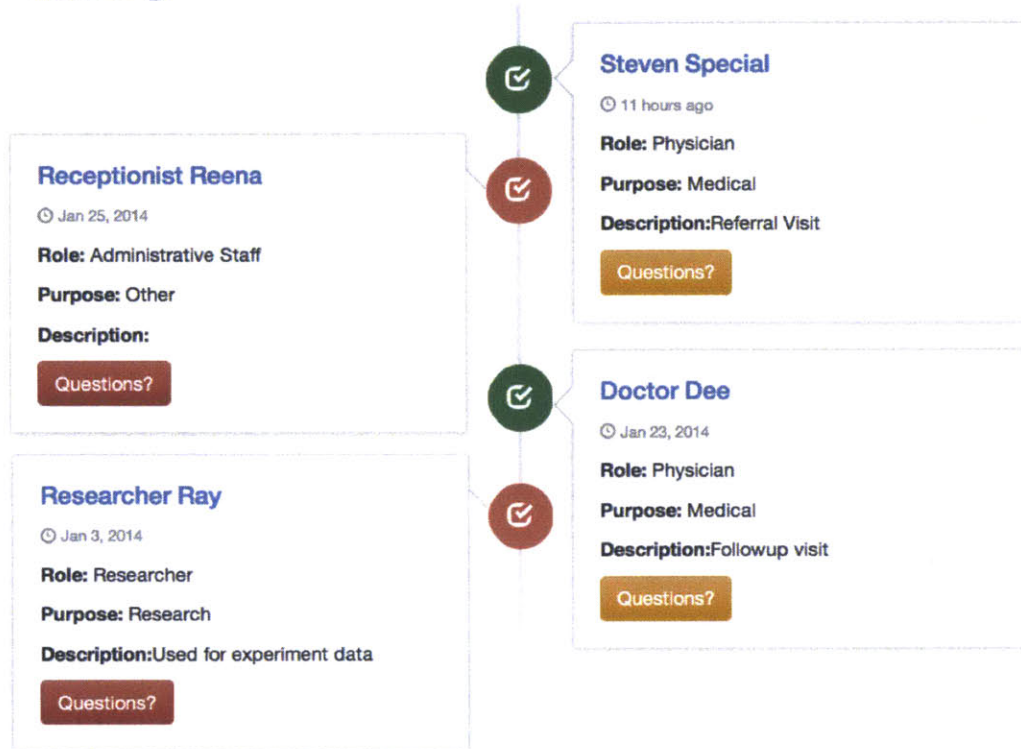


Figure 4-3: Audit Logs indicating how a sensitive data was used in Transparent Health. Depending on the usage restrictions set by the user, all the questionable usages appear on the left. The user can submit a question about the respective usage and subsequently flag the usage as a privacy violation.

4.5.1 Participant Profiles and Preliminary Questions

We recruited 25 participants for the study (17 male and 8 female, ages ranging from 18-55). All of these participants indicated that they visit a health care provider at least once a year with the median number of visits being 5. 20 participants indicated that they have access to their health care records after a visit to their doctor through an online health care portal. From the participants who indicated that they do not have access to their health data after a visit to the doctor, all but one expressed interest in using a system as such. Then we asked them if they are worried about their sensitive health information being misused in electronic health care record systems. 15 answered yes, 8 answered no, and 2 answered that they don't care. Then after-

wards, we gave some background as to how their private health data can be misused. Provided that there are means to figure out a privacy violation, we asked them what they consider most important in knowing: (1) Who? : the identity of the personnel that misused their information, (2) When? : the time at which the information misuse happened, (3) How? : how did they have access to the information and how they misused the information, (4) Where? : from where did they get access to the information, and where did they send the information to, (5) Why? : the motivations behind the data misuse, and (6) What did they misuse?. We specifically asked them to categorize their responses as **Rank 1**, **Rank 2** and **Rank 3**. The results are summarized in Figure 4-4. The results suggest that most users are interested in knowing ‘who’ misused the information, followed by ‘how’ the misuse happened, and ‘where’ the violators got access to the sensitive health information. This validated our implementation design decision of requiring agents in PETS to have a unified identity so that they could be identified in case of a violation. Also, the provenance trail is designed to provide enough evidence of other conditions in a misuse.

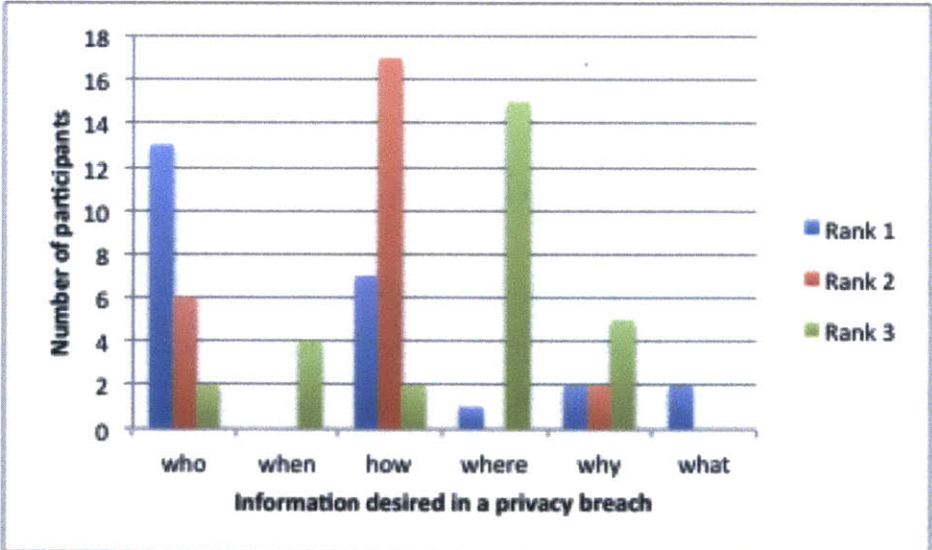


Figure 4-4: Categorization of what users consider most useful in knowing if there is a mechanism to figure out privacy breaches of their sensitive information

4.5.2 Creating the Health Profile

After interviewing the users, we asked them to try out our transparent personal health information system available at <http://www.transparent-health.us>. The first task they had to do was to create their health profile in the system without revealing their sensitive personal health data (nothing prevented them from entering their personal data, but we advised them not to add anything too personally revealing). We presumed that self created health profiles would give the participants a better sense of privacy awareness, rather than giving them canned health profiles that they are not able to identify with. To help the users with the process of creating the profile, we provided them with a health information profile creation guide. The guide suggested several medical conditions that they can choose from to add to the past and current medical conditions. The guide also provided the ability to add any other illnesses the users want in their profile. Based on the illnesses that were selected before, a medication list was provided. They had the option of removing some of the medications and adding some other medications. We also asked them the names of their primary care providers, and a specialist that they were referred to, to provide more identifying context for the user.

4.5.3 Setting Usage Restrictions

For each of the data item the participants entered, they had the option to mark that as a sensitive data item. An example would be to mark a health condition such as 'HIV AIDS' as sensitive. To do this they were presented with an interface to select (1) with whom they would like to share this information with (i.e. researchers, insurance companies, affiliates, and non-medical staff such as hospital receptionists, etc.) and (2) for what purposes (i.e. research, insurance claim processing, marketing, or other purposes). Please note that the physician, nurse and pharmacist roles in Transparent Health all have comprehensive access to the patient's health record by default, but they have to specify the purpose when accessing any data marked as sensitive.

4.5.4 Simulating Information Flows

Based on the information provided, we simulated several scenarios asking the users to acknowledge that the events in the scenarios happened. Examples of these events include: the doctor diagnosing one of the illnesses the user had picked, the user picking up the medications from the pharmacist, the doctor referring the user to a specialist, the participant agreeing to contribute the personal medical data for a research experiment by signing a waiver, etc. These events simulated some of the real world events that may have happened with the user knowing about them. As these events were being acknowledged by the user, the corresponding usage logs were generated and added to the PTN. We also added two other random events, the first event can be construed as a misuse of the patient's private health information (transfer of the medications information to a marketing firm by the pharmacist), and the other event is a treatment related activity that the user was not aware of (referral event where the doctor is sending the medical record to a specialist).

4.5.5 Auditing their Health Records

After the users finished completing their health profile, we asked them to test out the functionality of the 'Audit' button. Their task was to go to their health profile, and select the data fields they marked as sensitive and flag the privacy violations as identified by PETS are indeed privacy violations. An example audit log is given in Figure 4-3. PETS does a trivial inference based on the usage restrictions set on the sensitive data items by the participants to identify potential privacy violations. One of the random events we added, as described in the previous section, was designed to be a misuse of the data potentially leading to a privacy violation. 21 of the participants indicated that they like the feature of being able to see how their information was used by those who are authorized to work with their personal health information. When asked if they feel that the identified events are indeed privacy violations, 18 said yes, 3 said no, and the other participants said they don't mind if those agents viewed their data in that way.

4.5.6 Reversing the Roles

Next, we asked if they would agree to use a system such as Transparent Health if they were to take the roles of health workers that can be audited by the patients. 16 participants said yes, whereas the rest said that they would only use such a system, only if it was mandated by a law. Many participants indicated that if as health workers if there is nothing to hide, the patient has a right to the information.

4.5.7 Our Hypothesis and Supporting Anecdotes

It was our hypothesis that users will have a better understanding about their overall health care, and have a better confidence in electronic health care systems since they will be able to see if there has been any unwarranted accesses and usages of their protected private health information. Here are some of the anecdotes from the participants from the user study that supports our hypothesis: “A very innovative thought! This kind of site will be indispensable after few years.”, “Auditing my health information is easy from Transparent Health”, “It is a good system to ask questions from the doctors about my health information”.

4.6 Summary

Transparent Health, described in this chapter, was the first use case and validation of HTTPPA. The evaluation of the system was in the form of a user study that showed general acceptance of such a system. However, getting such a system underway in regular hospital environments can bring immense challenges. While patients directly benefit from seeing how their sensitive information flows within the hospital environments, there are no clear benefits for the hospitals to implement such a system. Furthermore, the hospitals may have fears of litigation from patients in the case of information breaches that can be revealed far too quickly. Therefore, this is an area that is ripe for regulation, and should governments decide to provide patients with a transparent view of sensitive information handling systems such as Transparent

Health that utilizes HTTPPA may be used. In the next chapter, I will describe PhotoRM and the accountable meme generator, the second use case of HTTPPA, in which users are encouraged to use and share copyrighted media appropriately.

Chapter 5

Use Case 2 - Accountable Systems for Intellectual Property Rights Protection

This chapter introduces another proof-of-concept application of HTTPA. We developed two HTTPA websites for photo sharing, and a browser based tool that can be used to add text on any photo on the Web to create an image macro to be shared on the Web. The two websites serve as image sharing sites similar to `imgur.com` and/or `flickr.com`. Users who have an interest in determining the virality of their content on the Web can contribute their content to these websites. Using the audit functionalities provided in the websites and the client-side tool, they can later determine who has used their images, and in what way. The browser based tool provides a mechanism for those who are interested in creating funny image captions to do so very easily, right from the browser, and share them on the accountable photo sharing websites provided or some other select popular photo sharing websites in a more accountable manner. We evaluated this end-to-end accountable system with a user study that indicated a lot of enthusiasm for this technology.

5.1 Accountable Client: Accountable Image Macro Generator

This section describes a client side implementation of HTTPPA that enables a user to modify an image that she finds while browsing the Web by adding some text on it. The tool displays the source of the image, and any licensing restrictions if available. The owner of an image is able to ‘audit’ an image to see how the image has been reused. The tool communicates with HTTPPA websites, as well as select non-HTTPPA websites.

5.1.1 Introduction

The generative culture on the web promotes the propagation of memes that often start out with image macros. An ‘image macro’ is a broad term used to describe captioned images that typically consist of a picture and a witty message or a catchphrase. Image macros or memes are becoming more numerous and popular every day. Someone stumbles upon a silly, cute, picture online. Often it is someone’s cat wearing an outfit, for example. The user who finds this image then modifies it in some way and shares it online through the various social media mechanisms that are available at her disposal. Users who find this manipulated image laugh and enjoy the strange image. But many then use the original template photo to make their own images, or modify the subsequent meme to say something a bit more funny, after which they post and share them with each other. Soon, the silly modified images gain massive popularity on social media outlets.

This scenario raises several interesting issues. Firstly, users who view the manipulated image may not be aware of the actual location of the image or any usage restrictions associated with it. If they want to reuse the image properly or avoid any copyright violations they should check back with the original location and modify the image as they see fit. So, in order to find the image and any usage restrictions on the image, they have to either rely on the information posted by the user or do their own

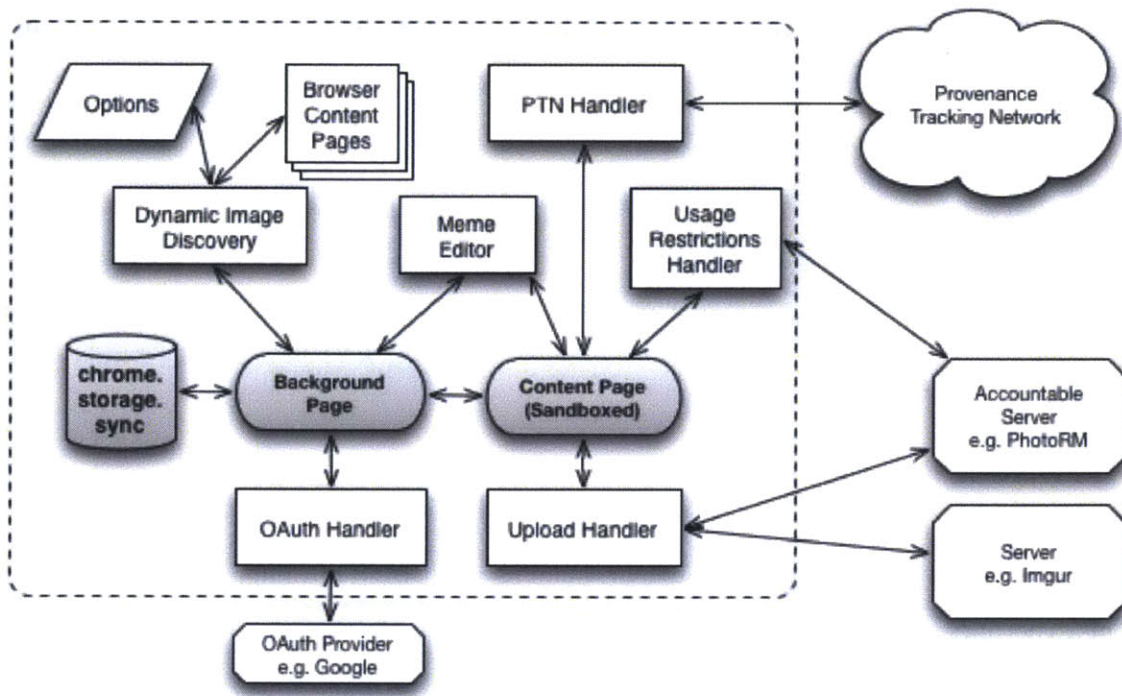


Figure 5-1: Architecture of the Accountable Client Side Tool

web sleuthing to find that out. Secondly, the original owners of the source image may want to find out the virality of the image, and any inappropriate uses of the image to request takedowns if necessary. If the original owners make all the photos private, they may be self-censoring and the potentially-public photos marked non-public may lose value, and would not be of any use to anyone.

Therefore, to address these two issues, we developed a browser based tool that uses HTTPPA to handle provenance and acceptable use. This tool displays information about the source image clearly to the user, allows setting usage restrictions or licenses, and enables sharing of the image seamlessly.

5.1.2 Implementation

The client side tool was implemented as a Chrome extension. Figure 5-1 displays the various components of the tool. Chrome specific components, i.e. the *sandboxed content page*, the *background page*, and *chrome storage*, are shaded in the diagram.

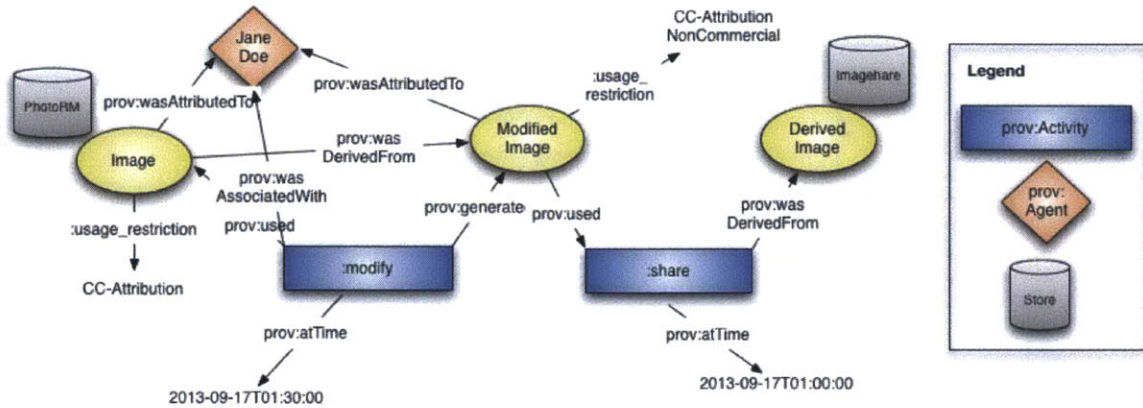


Figure 5-2: Provenance Information Flow in the Accountable Client Side Tool

Using the options specified by the user, such as what kind of images to detect based on the type and the dimensions, or the location serving that image, the *Dynamic Image Discovery* component will extract images from a web page and display them to the user for potential modification. This is all done through the background page that also handles one-time authentication of the user, in which the user is prompted for her credentials when the tool is first installed. Once a user has selected an image to modify, if the image is hosted in an Accountable Server, the content page consults the server for any usage restrictions for that image using the *Usage Restrictions Handler*, and the *PTN Handler* for the provenance information. The image is drawn on a canvas, and all the user actions are recorded from that. Once the user has finished modifying the image, the *content page* allows the user to set a usage restriction, afterwards the user may download it to local disk or share it on the Web. All of these actions are encapsulated in an RDF graph using Prov-O, and communicated to the PTN. For example, if the user modifies an image from PhotoRM.org and shares the image of Imagehare.com, a provenance graph similar to the one shown in Figure 5-2 is created and updated on the PTN.

5.1.3 Using the Tool

The tool is available for download and install from <http://tinyurl.com/accountable-client>. Figure 5-3 displays how it appears on the Chrome Web App Store. When

installing the extension, the user will be prompted for authentication with Google+ and will be requested some permissions, as the extension needs to communicate with the PTN to update the usage activities of the user. The options page as shown in Figure 5-4 gives the profile URL of the current user, so that the user can verify that she is correctly authenticated with the PTN.

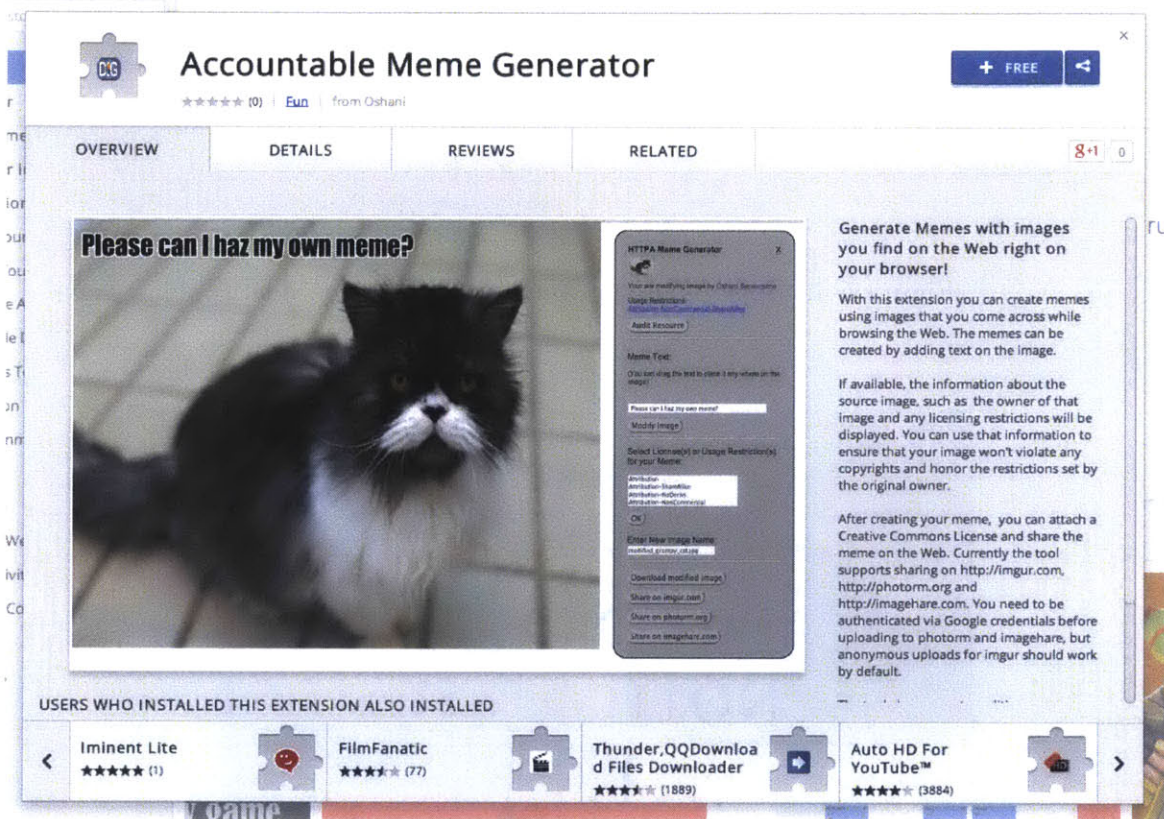


Figure 5-3: Accountable Client Side Tool on the Chrome Web Store

Serendipitous Discovery of Images

As the user browses the web, the browser extension displays a small icon in the address bar as shown in Figure 5-5 to notify the user of the availability of any images that she would want to modify. The user can customize what kind of images, the size and the source of the images they would like to be captured by the browser extension from the options page as shown in Figure 5-4.

Once the user clicks on the images displayed on the popup window or visit the

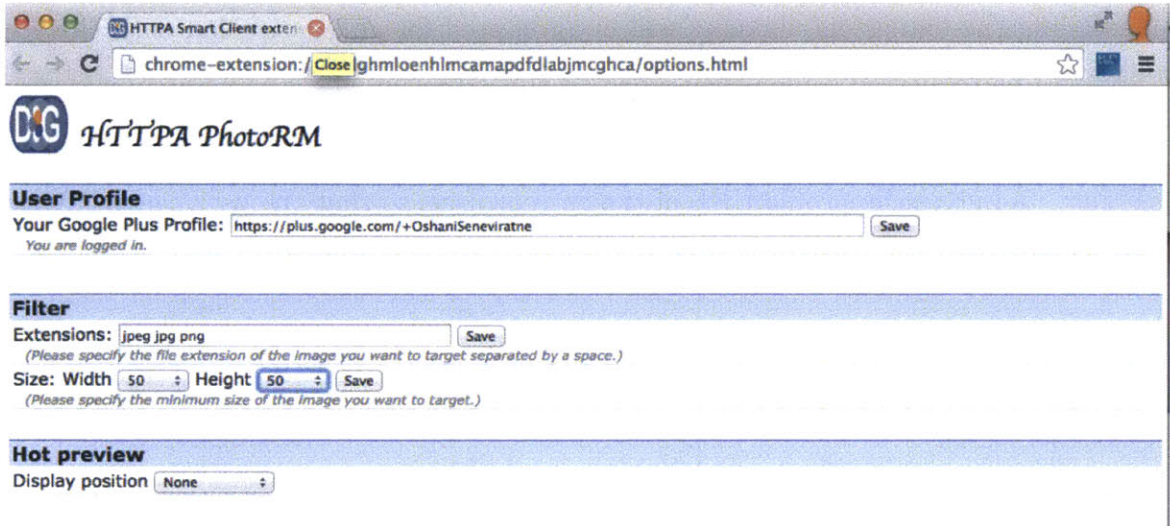


Figure 5-4: Options in the Accountable Client Side Tool

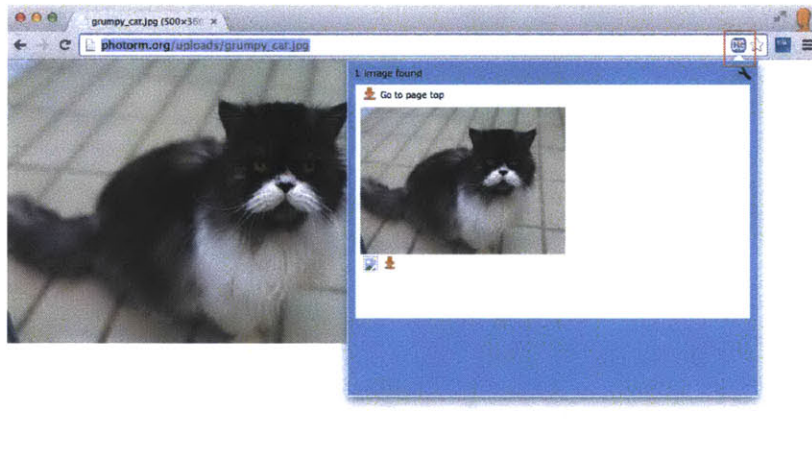


Figure 5-5: Using the Accountable Client Side Tool

image by entering the image URL on the browser, a grey sidebar will appear as shown in Figure 5-6. If available, the information about the source image, such as the owner of that image, any licensing restrictions, along with the derivation history for this image resource will be displayed (highlighted by the red box in Figure 5-6). The user can take that information into account to ensure that the modified image will not violate any copyrights and honor the usage restrictions set by the original owner.

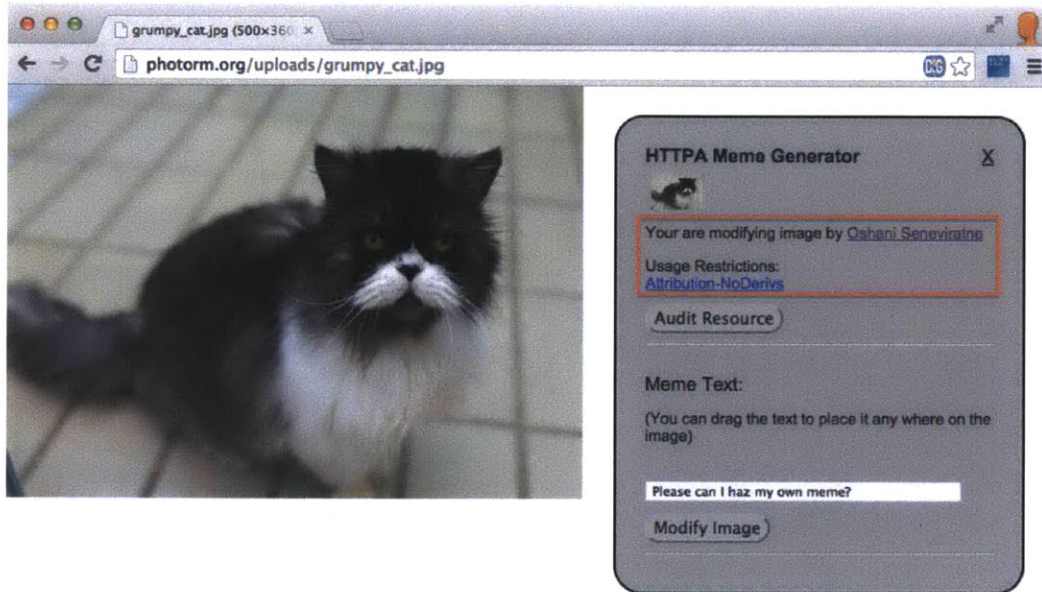


Figure 5-6: Image Macro Generator Before Editing Image

Creating The Image Macro

To create the image macro the user has to simply add some text and click the “Modify Image” button. The resulting image is shown in Figure 5-7. After creating the image macro, the user can attach a Creative Commons License. In this tool we discourage blanket strategies for usage restrictions, since the use of a single setting for all photos is often meaningless. Therefore, the tool allows setting a usage restriction or a CC license per image resource.

Sharing The Modified Image

After modifying and optionally attaching a usage restriction to the modified image, the user may download the image, or share the image on the free image hosting service <http://imgur.com> that does not support HTTPPA, and/or on the two accountable image sharing systems that support HTTPPA, namely: <http://photorm.org> and <http://imagehare.com>. The user need to be authenticated with Google credentials prior to upload to PhotoRM and ImageHare, but need not be authenticated on Imgur as the tool will upload the image as an anonymous image upload. Once the upload

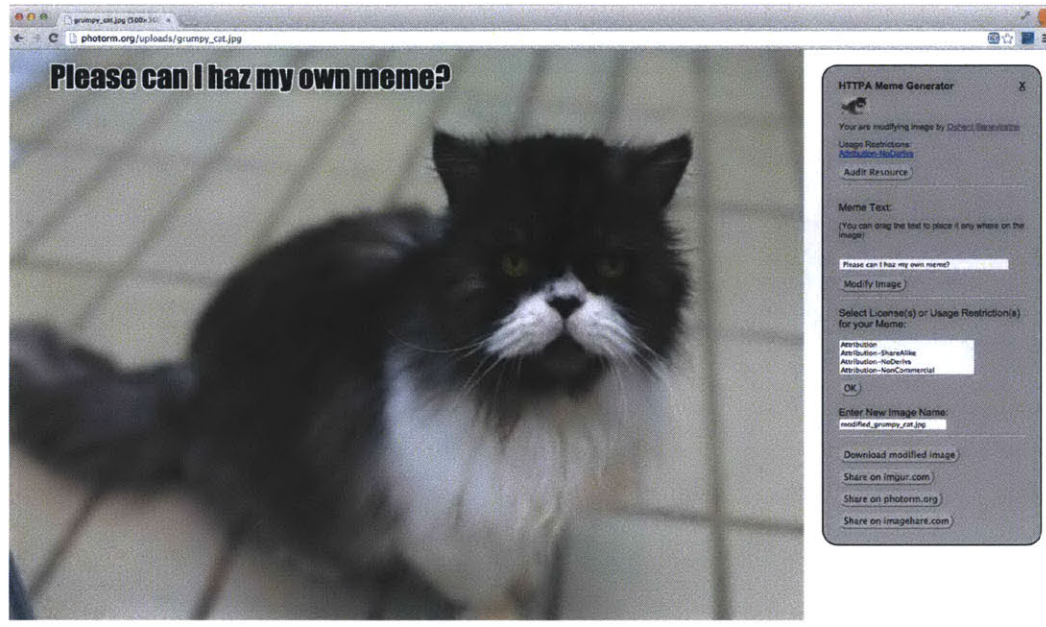


Figure 5-7: Image Macro Generator After Editing Image

on any of these websites has finished, the user will receive the location of the new (modified) image resource.


Auditing an Image Resource

Any resources that the user created can be ‘audited’ by clicking on the “Audit Resource”. Each image has a designated *prov:Agent* in the PTN, and if anyone other than that particular individual designated in the *prov:Agent* entry tries to access the audit log, the PTN will not return the audit log. Therefore the user must be authenticated via the tool before accessing the audit log. The audit log gives information as to who has accessed, downloaded and shared the images as seen in Figure 5-8. Since URIs of subsequent derivations are given, the user can check how the images have been used. Each log entry also lists the corresponding usage restrictions given to the derived image resource. HTTPA requires this to be a de-referenceable URI pointing to a description of the usage restriction that is machine and/or human readable. The identity of the re-user and the time of the activity are also given. The ‘?’ button next to each entry let’s the user clarify any questions about the reuse from the re-user,

which will subsequently be recorded in the PTN as well.

chrome-extension://dkblghmloenhlmcamapdfdlabjmcghca/audit.html

Audit Log for Photo



http://photorm.org/uploads/grumpy_cat.jpg

share by [Oshani Seneviratne](#) as <http://imagehare.com/uploads/birUp1aC.png> at 2014-08-03T16:33:53.449Z ?
with the following usage restriction(s):

- [Attribution](#)

share by [Oshani Seneviratne](#) as <http://photorm.org/uploads/MdKxJf29.png> at 2014-08-03T16:33:48.249Z ?
with the following usage restriction(s):

- [Attribution](#)

share by [Oshani Seneviratne](#) as <http://i.imgur.com/8i1JQBB.jpg> at 2014-08-03T16:33:42.668Z ?
with the following usage restriction(s):

- [Attribution](#)

download by [Oshani Seneviratne](#) as [modified_grumpy_cat.jpg](#) at 2014-08-03T16:33:37.187Z ?
with the following usage restriction(s):

- [Attribution](#)

share by [Jane Doe](#) as <http://i.imgur.com/Xbw0BHf.jpg> at 2014-08-03T03:15:52.993Z ?

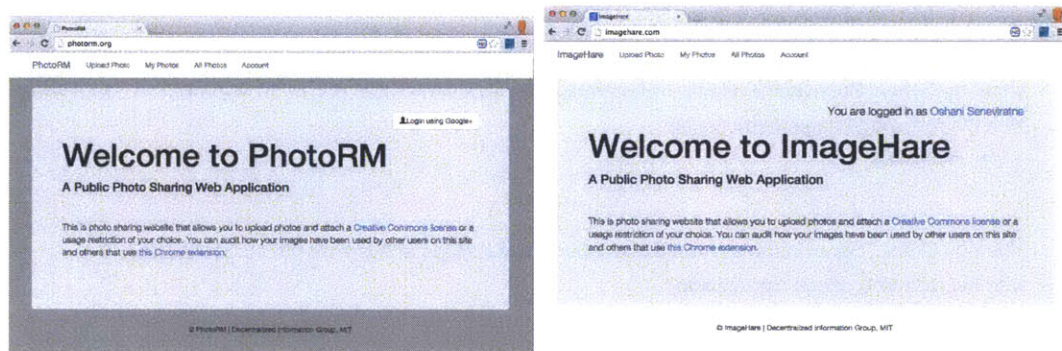
access by [Jane Doe](#)

Figure 5-8: Client-side Audit Log for a Resource

5.2 Complementary Accountable Server-Side Systems

We created two photo sharing websites, namely PhotoRM (<http://photorm.org>), and ImageHare (<http://imagehare.com>) that implement HTTPPA to supplement the client side tool described in the previous section. These websites require users to login with their Google+ profile credentials¹.

Once logged in, the users can upload images and attach a Creative Commons license or a usage restriction of their choice. These sites also provide an audit functionality to see the accesses and usages of the resource. Although the look and feel and the functionality of these two websites are identical, they are run on two completely separate servers, have their own data stores, and do not share any kind of information with each other. Figures 5-9(a) and 5-9(b) shows these two websites.



(a) PhotoRM.org (unauthenticated)

(b) ImageHare.com (authenticated)

Figure 5-9: Accountable Image Sharing Websites PhotoRM and ImageHare

5.2.1 Using the Accountable Photo Sharing Websites

Uploading an Image with a Specific Usage Restriction

Users of both PhotoRM and ImageHare are able to upload images and attach a usage restriction of their liking. Figure 5-10 shows the upload interface where they can

¹The choice of using Google+ was completely arbitrary and it was mainly due to the ease of use in recruiting participants for the user-study and the availability of the public profile URI on Google+. We note that any other single-sign-on authentication mechanism can be used here.

select a usage restriction as they upload the file. Although the usage restrictions that are displayed currently on the websites are Creative Commons licenses, this can be extended to other types of usage restrictions and licenses. Once the file uploads successfully, the website sends the information about the new website to the PTN.

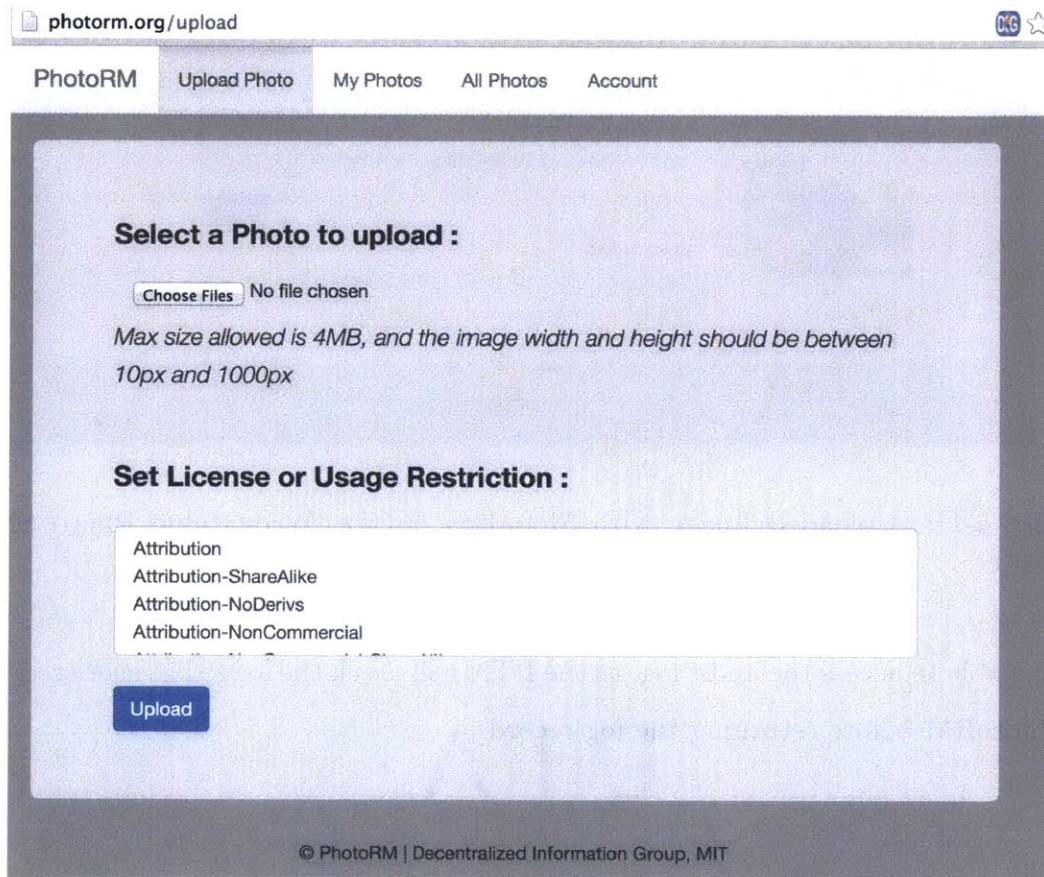


Figure 5-10: Upload Image in Accountable Photo Sharing Website

The uploaded images by the user are given in Figure 5-11. The information shown on the “Details” tab are acquired from the website’s own data store and displays certain properties of the image.

Auditing the Usages of Images uploaded

The “Audit” button in Figure 5-11 results in an audit log request for the corresponding image resource from the PTN. Assuming that the image was accessed and reused by others, the audit log will display something similar to Figure 5-12. Other users will

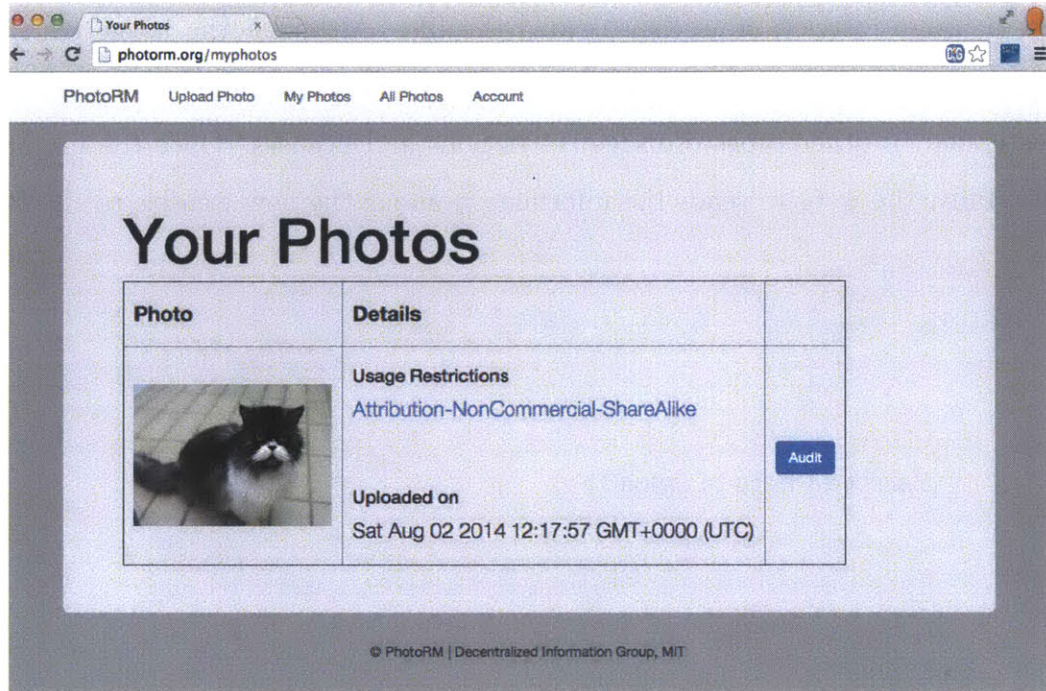


Figure 5-11: Uploaded Image with Metadata on the Accountable Photo Sharing Website

not be able to access the audit log, as the PTN will check the logged in user credentials in PhotoRM before returning the log record.

5.3 Evaluation

We conducted a user study with the end-to-end accountable photo sharing setup comprising of the Accountable Client-side Tool and the Accountable Server-side implementations described in the previous sections. The goal of this experiment was to compare and contrast the reuse behavior of individuals who have access to accountability tools with those who did not have such tools available. Our hypothesis was that the users who had accountability tools are more attuned to the usage restrictions before modifying or sharing images, and similarly, the contributors of the images will be incentivized to share their images freely knowing that they can audit the usages of their images any time.

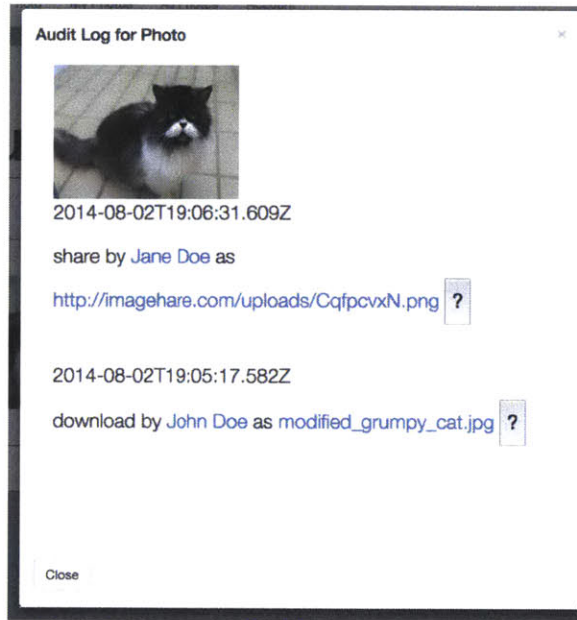


Figure 5-12: Audit Log for the Image Resource in PhotoRM

5.3.1 Methodology

This experiment had two parts. In the first part we gathered photos from participants, and in the second part we requested another set of participants to reuse those images in a creative manner with and without using Accountable Tools developed as part of this thesis. All the participants in this study were recruited from Amazon Mechanical Turk. To participate in the study, we required they have 98% approval rating, and have completed at least 5000 Human Intelligence Tasks on Amazon Mechanical Turk.

Part 1: Gathering Photos

We requested 56 individuals to contribute images on one of the HTTPPA based Photo Sharing websites (photorm.org). Before they each uploaded the image, we asked several questions to ascertain their previous photo sharing behavior. These questions include:

- Where have you shared photos (both public and private) on the Web?
- Have you set any privacy settings, usage restrictions or licenses on any of the images you have shared on the Web?

- What kind of access and usage restrictions have you used on your images?, i.e. Audience based (who is allowed to access and modify), Location based (images are only available in a certain geographical location), Re-share based (whether they allow sharing), Reuse based (what others can do with the images)
- Imagine you are a creative content creator (i.e. photographer, artist), and you have an image that you wish to share with others. How would you share it?
- If you had to give your own usage restrictions or licenses on a “creative image” upon sharing on the Web, what would those be?
- Are you interested in knowing how an image you shared on the Web was reused and re- shared?

We then asked them to upload an image and attach a suitable usage restriction or license. We gave the following options from Creative Commons:

- Attribution (mention my name, source, etc)
- Attribution-ShareAlike (when sharing use the same license)
- Attribution-NoDerivs (do not modify this image to make derivatives)
- Attribution-NonCommercial (do not use this image in a commercial setting)
- Attribution-NonCommercial-ShareAlike
- Attribution-NonCommercial-NoDerivs
- None of the above

Afterwards we gave them an audit log similar to the one shown in Figure 5-8 specifying some accesses, reuses and shares. We asked if the audit log indicates any activity that violates their original license or usage restriction on the image. We grouped their responses according to the original usage restriction they set on the image. We asked the participants to visit the location the image was reused at and determine if there was a violation. Our expectation was that they will check for

any attribution information, the license attached to the reuse image, and the types of modifications done on the image. For those individuals who set their preference to be Attribution (BY), Attribution-ShareAlike (BY-SA), no further actions were necessary. However, for those individuals who set any of the NonCommercial licenses, we inquired how they can use the current web technologies to determine if the photos they contribute are used for any commercial purposes or not. Finally we asked them what they would do if they find out if someone has reused their images on the Web in a manner that violates the original usage restrictions.

Part 2: Reusing Photos

In this part of the study, we recruited 105 participants and inquired about their previous image reuse behavior, asked them to add some text on an image to create an image macro, and asked them to do the same using the Accountable Systems developed as part of this thesis. Before they started creating the image macro we asked few questions about their prior image reuse behavior, such as:

- Have you ever taken an image from somewhere on the Web and modified it for a meme or for any other purpose?
- When reusing have you checked to see if the image had any copyright or licensing terms?
- Are you aware of any browser-based tools that let you modify images without downloading to your computer?
- If you have modified an image found on the Web, did you ultimately share it elsewhere on the Web?

We then gave them a random image uploaded by the participants in the Part 1 of the study, asked them to add some text on it and upload it to Imagehare.com. Afterwards, we asked them to install the Accountable Meme Generator chrome extension described in Section 5.1, and to create a similar image macro using the tool. When

using the tool we asked general questions about the usefulness of the provenance information displayed.

5.3.2 Results

Part 1: Gathering Photos

The participants who were asked to contribute an image mentioned that they have used Facebook (31%), Google Drive (14%), Imgur (13%) and Dropbox (13%) to share their images with others. Out of those who have shared images in the past, 75% (42 out of 56) have set some kind of privacy or usage restriction setting on the images shared. Many of these restrictions are audience based (53%), followed by reuse based (28%). An overwhelming 89% indicated that they are interested in knowing how an image they shared on the Web was reused. Many of them indicated that they do not know of any tools to let them check for reuse, while some others indicated that they can check on Facebook to see how their friends have shared their images. When uploading the image and attaching the usage restrictions, a majority of them (24%) specified the Attribution-NonCommercial-NoDerivs license.

Given the audit log similar to the one shown in Figure 5-8, the responses for individuals who set the different licenses are given in Figure 5-13. The x-axis indicates the license they specified, and the y-axis specifies the number of participants who thought the audit log indicated an instance where their usage restrictions were violated or not, or that they are unsure about the violation. As can be seen from Figure 5-13, Attribution-NonCommercial-NoDerivs (BY-NC-ND) was the most popular license, and a majority of them quite correctly indicated that the audit log presents a violation of their usage restriction, as it includes several derivatives and instances where there are no clear attribution information.

When asked about what they would do when they find out that someone has violated their usage restrictions, 19% of the participants indicated that they will find out the identity of the individuals who violated the usage restrictions, 14% indicated they will ask them to give proper attribution, 11% indicated they will give a takedown

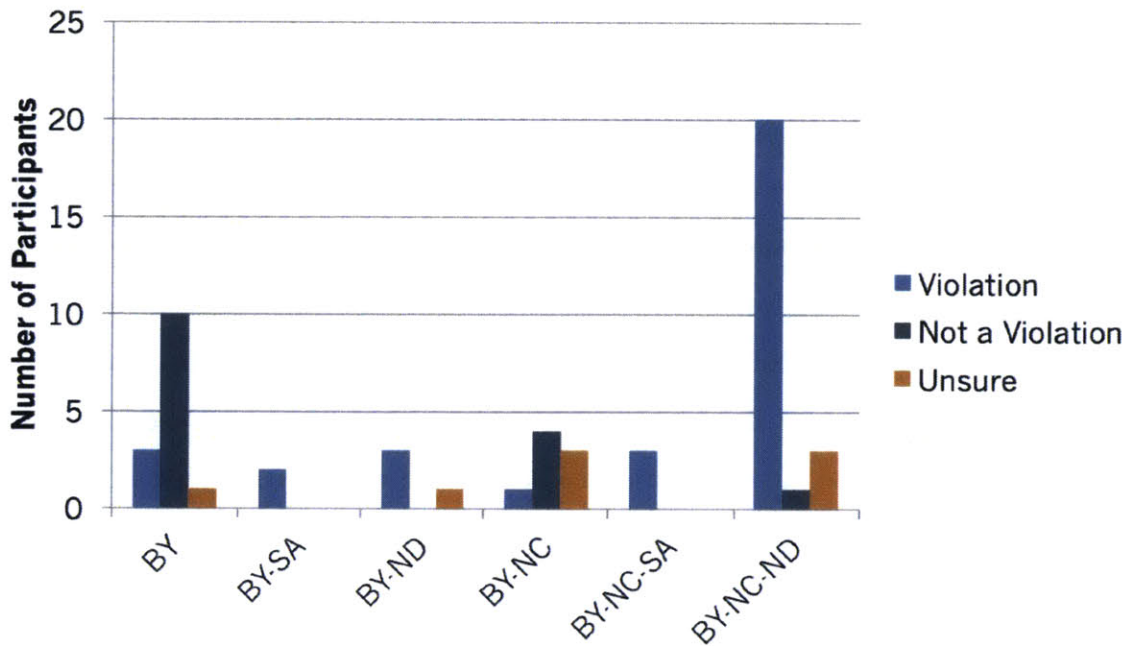


Figure 5-13: Perception of Study Participants as to the Potential Violation of Usage Restrictions given in the Audit Log

notice, 12% will request monetary compensation, 13% will give a warning and ask the violators not to do the same thing again, and 9% will tighten existing usage restrictions in the future based on this experience.

The HTTPPA based accountable tools provides mechanisms to determine the source information, the derivation of a given resource, and the identity of the individuals who reused the content. Using the Accountable Meme Generator, participants of this study were able to ascertain if someone had violated their usage restrictions, and provided a mechanism to contact them via their public profile page or through email to take action in the case of any usage restriction breaches. Similarly, the re-users are empowered with information that helps them do the right thing.

Part 2: Reusing Photos

Prior to completing the tasks, 60% of the participants indicated that they have taken an image from somewhere on the Web and modified it previously in order to create an image macro or for some other similar purpose. However, only 19% of them have

checked to see if the image they had taken had any copyright or licensing terms attached with it. 84% of the participants indicated they are not aware of any browser based tool for modifying images without downloading, and the participants who indicated that they are aware of such tools only gave pointers to image editing websites, and not browser based tools. Out of those who have modified the image, 54% of them have shared that image, for example on Facebook for the enjoyment of their friends.

There was a marked difference in user behavior when modifying the image without the Accountability tool (Case 1) and with the Accountability tool (Case 2). Figure 5-14 shows the number of participants in both these cases with respect to checking usage restrictions, modifying the given image, and sharing the modified image. For Case 1, we gathered this information by asking the users, where as for Case 2, we logged the user actions via the tool to learn their behavior. As can be seen from Figure 5-14, the Accountable Meme Generator provides a direct mechanism to let users check usage restrictions for the image resource. Given that a majority of the images collected from Part 1 of the study had Attribution-NonCommercial-NoDerivs (BY-NC-ND) licenses, those who were using the tool did not modify or share the image although in the earlier case they did so due to the lack of awareness of the licensing and provenance information associated with the image.

After modifying the images, 90% of the participants claimed to have seen the information about the owner of the image and the license, and 88% of them said it was very useful to have given the task they were assigned to do in the study. 86% preferred the auditing feature that allows a user to trace how her images are used on the Web, and 80% indicated that they would be willing to reveal their identity should such a mechanism is adopted on the vast majority of websites.

Anecdotes From Participants

Here are some anecdotes from the participants that support the ideas of Accountable Systems. “I like how I can keep track of how many times an image of mine has been downloaded and by whom”. “It seems a little overkill to me. But it might be sort of fun to share some of my more important images and see what the world does with

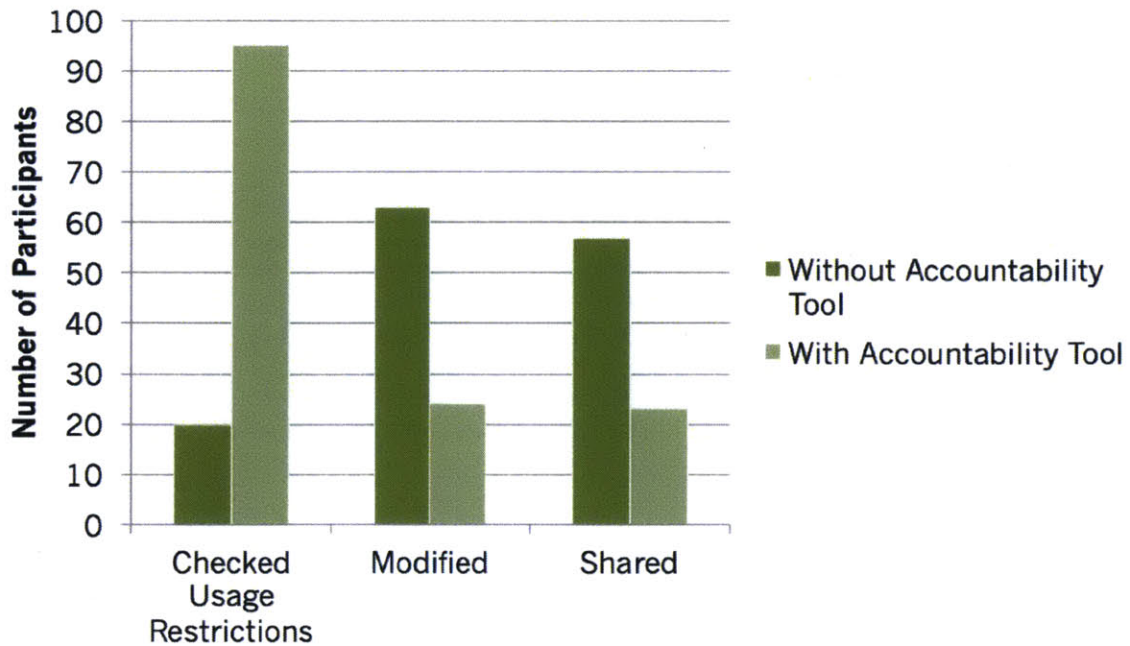


Figure 5-14: Reuse Behavior of Individuals with and without the accountability tool

them!” “It looks like a good idea for artists and photographers.” “Interesting idea. Could see this being useful for finding art references/stock art for personal projects.”

5.4 Summary

PhotoRM, ImageHare and Accountable Meme Generator, described in this chapter combined together comprises the second use case and validation of HTTPPA. The user study showed a strong acceptance of these types of tools. Initially when the users did not use the Accountable Meme Generator, it resulted in less compliance to the stated usage restrictions on the photos. Many of the photo contributors preferred the audit capability offered by the HTTPPA aware websites as opposed to traditional photo sharing websites, and some of them indicated that their perceptions for attaching usage restrictions changed after seeing how their photos were used. In the next chapter, I present some related work in addressing privacy, intellectual property rights violations and technologies that enable provenance.

Chapter 6

Related Work

In this chapter, I examine some of the popular technologies on privacy protection and intellectual property rights protection as well as technologies that provide provenance on the Web. These technologies are then compared and contrasted them with the technologies that drive HTTPPA based accountable systems infrastructure.

6.1 Privacy Protection

6.1.1 Platform for Privacy Preferences (P3P)

Platform for Privacy Preferences (P3P) protocol was developed at the W3C with the intention of communicating the privacy policies of websites to user-agents who connect with them [84]. The P3P recommendation allows website operators to express their data collection, use, sharing, and retention practices in a machine-readable format. The idea was for tools to be built to compare each policy against the user's privacy preferences and assist the user in deciding when to exchange data with the websites. For example, a user-agent such as a browser, can retrieve a machine readable privacy policy from the web server from a well-known location and respond appropriately (for e.g. display symbols or prompt the user for action). Privacy Bird was one such implementation [85]. In the cases where the policy file is not available at a well-known location in the site, the server must declare the location of the policy reference file

by using a special HTTP header or by embedding a LINK tag in the HTML files to which the P3P policies apply. P3P user agents typically allow users to specify their privacy preferences so that they can automatically compare a website's policies to their preferences.

Even though P3P became a W3C recommendation, it has several limitations: a complicated language to express policies, the inability to express preferences on third party data collection, or multiple privacy policies for one web page [86], and the lack of alignment on incentives for adoption. These limitations have prevented P3P from wide adoption.

HTTPPA builds on the notions introduced by the P3P, but instead of a complicated language there is support for simple, extensible, domain-dependent machine readable vocabulary in defining privacy choices. For the HTTPPA protocol to work, both the clients and the servers should work together. In P3P the client can ignore the stated privacy policy, and the servers can give a fake privacy policy just to be able to serve clients that demand a P3P policy. HTTPPA has the added advantage for content providers where they can see how the content was reused, thus more incentives for adoption compared to P3P where the privacy notice and how it has been honored are completely opaque to the content provider.

6.1.2 Notice and Choice Models

There are many attempts at providing information about appropriate use of content and information. Some of these those technologies are described below.

The W3C POWDER (Protocol for Web Description Resources) language provides a mechanism for describing groups of resources by essentially grouping URIs and linking these groups of URIs to a group of common XML statements regarding topics like authentication [87]. While more generic than P3P, it is aimed at similar privacy use-cases such as privacy descriptions for child protection. While it is useful to describe attributes associated with groups of URIs rather than single URIs, it is seen as complex and has failed to gain deployment for the same reasons as P3P. Unlike POWDER, HTTPPA provides de-referenceable URIs that points to the usage restric-

tions and the intentions. Also, since the usage restrictions are expressed in RDF, it is more expressive than the POWDER descriptions in XML.

Mozilla Privacy Icons takes a simple icon-based approach inspired by the Creative Commons [88]. Instead of specifying every possible type of privacy and data-handling scenario, they specify only a few common privacy scenarios that users can encounter: such as information sharing, storage, monetization, deletion and contact/notification. The icons are designed to be easy to use and be understood by ordinary end-users. As online businesses are looking for ways to build trust and manage consumer expectations through transparency, choice, and accountability, these privacy icons seem to be a timely solution. However, since it is detrimental for sites that violate user privacy to label themselves as such, it would be up to the browser or a browser app to automatically label such sites. Also, users do not ordinarily notice an icon by its absence but only by its presence. Therefore the browser/app should detect the absence of the privacy icons to notify users they have entered a site where their privacy and usage restrictions could be violated.

Primelife's "D. Dashboard" provides an adequate notice to the user by making a limited assessment of the current web page and creating an icon based upon factors such as the use of 3rd party cookies, the use of P3P, etc [89]. The Dashboard is available in the form of an extension that logs the user's HTTP traffic to a local database and provides a variety of queries for analyzing them.

IETF's GeoPriv proposal puts privacy policies in the hands of users instead of services, where a user transmits her own privacy preferences about how her location data should be used, while the websites are bound by their market or legal obligations to respect those preferences [90]. W3C's Geolocation API also advocates that websites disclose their data usage practices to the user [91], although it is rarely practiced by most websites that implement the API [92].

The Simple Policy Negotiation for Location Disclosure proposal describes a system that lets a user have a dialogue with a website that uses her location data before disclosure [93].

HTTPPA based client side tools provide similar functionality to these tools and

techniques, by displaying the provenance information along with any appropriate use information to help guide the user when reusing content or information.

6.1.3 Transparency Enhancing Tools (TETs)

One of the products of Accountable Systems is to provide a transparent view of the information usage to a subject. Transparency in general can be in conflict with the privacy principle of data minimization. The goal of transparency is to make information available, while the goal of data minimization is to minimize the amount of information available.

Pulls et al have proposed a privacy preserving TET that uses a cryptographic system for enabling privacy logging and for storing the sensitive data [94]. Their cryptographic scheme makes it impossible to link multiple log entries for the same user or user identifiers across multiple data processors. For example, their system can make access to electronic health records transparent to the patients to whom the records relate, while preserving the privacy by not revealing the same information to another unauthorized individual. Their transparency scheme differs to that of HTTPPA at the point where they do not allow an unauthorized party to access the information. HTTPPA does not stipulate any enforcement on who can access and what, but rather enable access to such information within a given domain as long as they are authenticated.

Sackmann et al. discuss an approach based on what they call privacy evidence [95]. The key components in this system is a secure logging facility and an automated privacy audit component to give the user information on how well a system fulfills the promised (or user provided) privacy policy. The main difference in this system with HTTPPA is that it is implemented as a centralized system whereas HTTPPA is not.

PeerTrust provides a mechanism for gaining access to secure information or services on the web by using semantic annotations, policies and automated trust negotiation [96]. In PeerTrust, trust is established incrementally through an iterative process which involves gradually disclosing credentials and requests for credentials.

PeerTrust is an interesting approach that expects both parties to exchange credentials in order to trust each other and assumes that policies are private. In HTTPPA too, there is a transparent communication of accepted use of the information disclosed prior to the information transfer. However, PeerTrust is only a part of the puzzle whereas HTTPPA is a comprehensive end-to-end architecture.

6.1.4 Privacy Protection in Electronic Healthcare Records (EHR) Systems

The need for technical solutions for maintaining the privacy of EHR systems is motivated in the literature [97, 98, 99, 100]. Many of these proposals argue for tighter sensitive information handling practices implemented through information security mechanisms. However there have been some work on providing transparent and accountable data access in health organizations as described in [101]. Their approach is to give unrestricted access to legitimate users and channel usage inquiries and usage justifications through an information accountability service. Unlike in our HTTPPA based Transparent Health system where the focus is on enabling a decentralized architecture to find breaches across systems and notify them to the data owner, their focus is on implementing a centralized architecture for individual health care providers. The patients in their systems can seek redress within the system and penalize misbehaving actors by restricting access to the records.

6.2 Intellectual Property Rights Protection

Some of the more popular ways of enabling intellectual property rights is through Digital Rights Management techniques, that includes watermarking, fingerprinting and various other encryption mechanisms [102]. However there are less restrictive approaches to protecting intellectual property rights that are comparable to Accountable Systems. These approaches are described below.

6.2.1 Making Attribution Easy

The Commons Machinery project [103] is building the architecture necessary to safeguard information such as attribution, terms of use, where the thing was created and persistently associate a creator with her creation in content management systems. Their approach is similar to some of the tools, i.e. the Semantic Clipboard, I developed during the early experimentation stage as described in Chapter 2. HTTPPA, not only makes attribution easy, but also enables the content creator to determine any inappropriate uses of her creative works that is not provided in the Commons Machinery project.

6.2.2 Allowing Fair Use

Project DReam describes an architecture where users can use Digital Rights Management (DRM) to control use of content under fair use terms [104]. The system they describe requires the user to connect to an anonymizing agent for authentication and assert fair use on any of the user-owned content. A user interface on the DRM software that is used to manage the content will ask the user to enter whether the reuse is for review purposes, educational uses, parody, or for other purposes. It will also ask the jurisdiction in which the content will be reused. The anonymizing agent will relay this information to the copyright owner for auditing. While there are some similarities of HTTPPA with the architecture proposed in Project DReaM, HTTPPA is designed to be much more general. In particular, HTTPPA handles any type of content, not just copyrighted material.

6.2.3 Reuse Detection

Reuse detection is important in domains such as plagiarism detection and even in biological sequence mining. Significant research has been carried out to detect reuse of text. This includes information retrieval techniques as mentioned in [105, 106], where the document is treated as a sequence of symbols and substring based fingerprints are extracted from the document to determine repetitive patterns. The approach used in

HTTPPA for reuse detection is logging, and if necessary, reasoning of these logs with rules about proper usage of this content is described in machine readable rules.

6.3 Provenance

There has been decades of research on enabling provenance in scientific workflows, but up until recently, provenance concepts have not been applied in relation to accountability. Many provenance systems generate provenance data locally and submit them to a central provenance management service. Although this provides consistency of the provenance data and provides an easy interface to query them, when the number of systems contributing to the provenance data increases, a centralized provenance log server may not be optimal in managing provenance information. In this section I will outline some of the related applications of provenance and how they relate to provenance mechanisms used in the PTN.

6.3.1 Lineage of Data

Provenance is traditionally used to prove lineage of data in scientific computing [107, 108], scientific workflows [109], on time-stamping a digital document [110], and more recently to provide meta-data about the origin of data [111]. All these works differ from HTTPPA because the provenance generation and maintenance in HTTPPA is delegated to a distributed network, in which provenance is kept in a centralized database.

Groth et al. [112] derive several principles for documentation of proper evidence of past execution. They mention that (i) Assertions must be based on observations made by software components, according to the principle of data locality; (ii) Each assertion making up documentation of the past for a computer system must be attributable to a particular software component. Using those principles outlined, provenance management tools such as 'ProvToolBox' [113] creates Java representations of the prov data model and enables manipulation of it from the Java programming language. ProvenanceJS [114] is a Javascript tool that can embed and extract provenance from

HTML pages. Both these tools are very document centric. HTTPPA handles provenance in a more generalized manner where it is applied in the context of tracing appropriate use of information across different websites.

6.3.2 Structured Logging

Structured logging, i.e. generating log files that incorporate dependencies between entities, the start and stop times of activities, and the inputs/outputs of activities, has been a topic of interest in many programming languages research [115]. There are many infrastructures, specifically in web service adaptors, that can be repurposed for collecting provenance [116]. Samavi et al describe a framework designed to facilitate privacy auditing through the use of two ontologies, whereby one provides provenance enabled logging of events, and the other for synthesizing the contextual integrity for compliance and observation derivation [117]. In comparison to our architecture, these systems have not implemented an end-to-end infrastructure such as the PTN in tracking privacy violations.

6.3.3 Semantic Web for Provenance

McGuinness and Pinheiro da Silva introduce Inference Web, an extension of the semantic web which aims to enable applications to generate portable explanations for any of their answers [118]. A key component in Inference Web is PML (Proof Markup Language) [119] that includes support for knowledge provenance and reasoning information. PML includes metadata such as authorship and authoritativeness of a source, but also reasoner assumptions (e.g. closed vs open world, naming assumption) and a detailed trace of inference rules applied (with variable bindings). Human-readable explanations are derived from the PML, and browsable representations can also be exposed to the user. In HTTPPA, provenance information is modeled using Prov-O, and the user actions are represented in a human readable format using the Accountable Clients.

6.3.4 Inline Provenance using Metadata

There are various technologies that allow one to transfer metadata along with the content by embedding the metadata in machine readable RDF. Extensible Metadata Platform (XMP) [120] is one such example. It is widely deployed in embedding licenses in free-floating multimedia content such as images, audio and video on the Web. Another format which is nearly universal when it comes to images is the Exchangeable Image File format (EXIF) [121]. The International Press Telecommunications Council (IPTC) photo metadata standard [122] is also another well known standard.

The metadata tags defined in these standards cover a broad spectrum including date & time information, camera settings, thumbnail for previews and more importantly, the description of the photos including the copyright information. One major drawback of inline metadata formats such as XMP and EXIF is that it is embedded in a binary file, completely opaque to nearly all users. In addition to that, these metadata formats can only handle limited number of properties and lack the rich expressivity offered by usage restriction specification using RDF offered by HTTPPA. Furthermore, in order to provide a more general accountable architecture for the Web, and to be media-agnostic, the provenance information in HTTPPA is stored separately from the content.

6.4 Summary

This chapter presented a discussion of related work from the problem domains of privacy, intellectual property rights protection and enabling provenance in systems. These works are compared and contrasted with the technologies utilized in realizing an Accountable Architecture on the Web with the HTTPPA protocol and the Provenance Tracking Network. In the next chapter I conclude this thesis with a summary of the work done, contributions, limitations of the architecture, and some future work to move the project forward.

Chapter 7

Conclusion

This chapter summarizes the contributions of this thesis, identifies some of the limitations of the proposed architecture, discusses some possible deployment strategies, and presents some future work that could realize the promise of Accountable Systems.

7.1 Thesis Summary

This thesis has addressed the problem of enabling and determining appropriate use of information on the Web with respect to privacy and intellectual property rights violations. We have seen the need for tools, techniques and standards that strike an appropriate balance between the rights which are held on the document and the power of reuse. The rights held on the document can be preserved by expressing what constitutes appropriate uses and restrictions using a rights expression language. Reuse can be simplified by providing the necessary tools that leverage these machine readable rights to make the users more aware of the usage restriction options available and ensure that the user be compliant. Such techniques can be incorporated with principles of provenance to achieve an end-to-end accountability architecture.

Accountable Systems makes transparency a first class citizen in information systems. This enables the information owner to check how her information has been used. The usage data can further be reasoned with individual, organizational, state or federal policies or usage restrictions to assert that no violation has taken place.

Therefore, the information producers will have more trust in the Accountable Systems while the information consumers will act appropriately and be less likely to misuse information. Thus, one of the main goals realized in Accountable Systems is that users are aware of, and are in control of their information on the Web. Another parallel goal of Accountable Systems is to let content contributors see how their content has been used, as well as enable content consumers to appropriately use these content.

I have defined Accountable Systems, built them, tested them at small scale, and shown that by supplementing the traditional access control mechanisms increases responsible re-use of sensitive and creative data, which in turn benefits both the consumer and the producer. A summary of research contributions from this thesis is given in Figure 7-1.

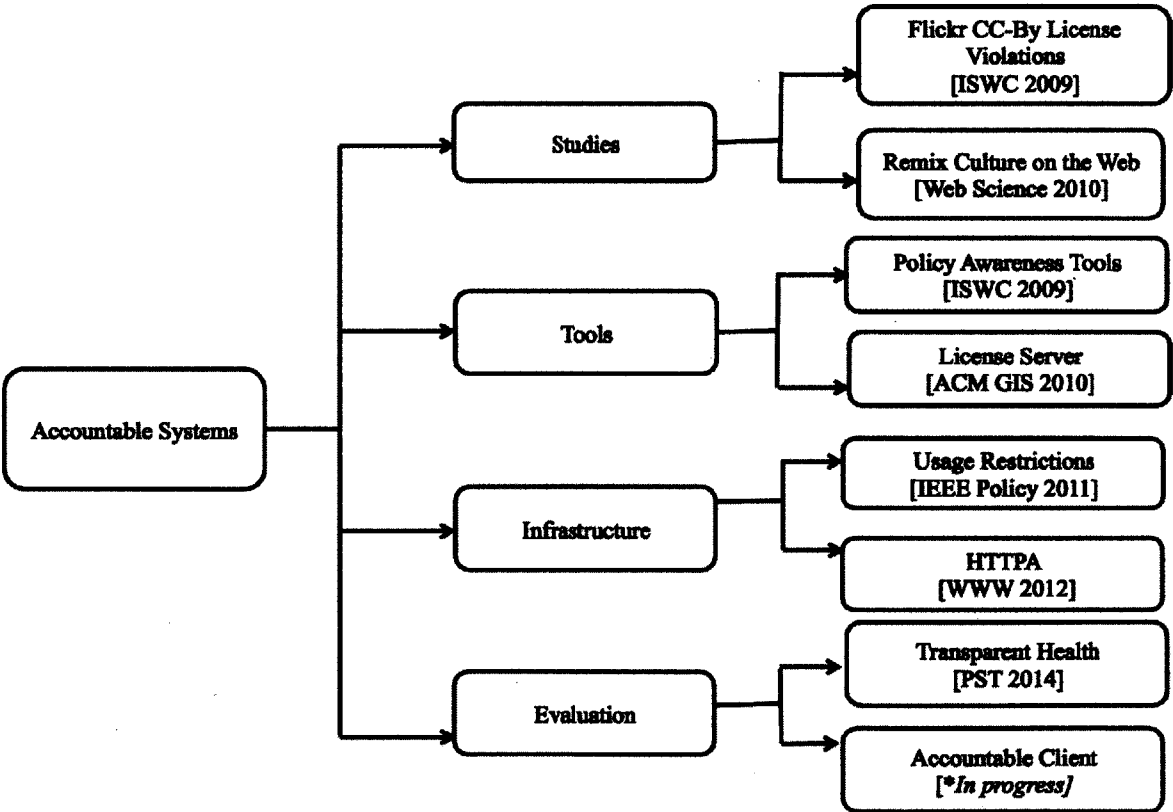


Figure 7-1: Research Contributions to Computer Science

7.2 Deployment Strategies

There are many social and economical hurdles that must be addressed before HTTPPA can be successfully deployed on the Web. In this section I present some strategies for HTTPPA to successfully transition from a research project to a real-world deployment.

7.2.1 Consortiums to Maintain PTNs

Running and maintaining a PTN node requires time, money and effort. The incentive for running a PTN node for a certain organization should match the benefits it derives from the PTN. Some of the early adopter communities that will find this form of accountability in their information handling practices will have specific needs. For example, a hospital will require secure storage of accountability logs in the PTN and would be more comfortable in taking part in a PTN that is run by a group of hospitals that can provide some form of information security assurances. Similarly, a company may not be willing to let the data usages in their company be recorded on a competitor's PTN node, or a company may not want its PTN resources utilized in storing the competitor's records that incurs a cost to the company. Such social and economical constraints may seem forbidding initially, but there are many examples in which similar minded groups have collaborated in achieving a common goal through consortiums that result in benefits for everybody. PlanetLab Consortium is one such example [123]. This is a collection of academic, industrial and government institutions that are cooperating to oversee the long-term growth of the PlanetLab network, maintenance of the hardware infrastructure, support and development of the software overlay, and defining policies that govern appropriate use.

By running the PTNs tailored to their needs, a consortium member can ensure that the nodes that join the network do so in good faith and will not be a rogue node. Every such node that joins can be endorsed by the authoritative body of the community that will be allowed to run the gateway nodes. Each PTN consortium will have its own acceptable use policies for connecting to the nodes. We envision that there will be a power law distribution with few larger PTN consortiums such as the ones listed below

(Healthcare Providers, Scientific Data Providers, Social Media Websites, etc.), and small PTNs in small communities (tennis clubs, churches, independent user groups, etc). In fact, we believe that at least in the short term, a world with many competing PTNs with different terms of service, mission statements, and membership models is healthier compared to a single global PTN.

Some of the proposed larger PTN consortiums are given below:

Healthcare Providers

Healthcare providers can be part of something similar to the Health Data Consortium [124], which is a public-private partnership that strives upon innovative use of health data to improve the healthcare industry. By participating in the healthcare PTN, the consortium members will have access to collective knowledge about the usage patterns of the health data. Additionally, they will be able to reap the benefits of the PTN: one such benefit is the ability to provide transparent view of health data to the patient. Additionally, the log records stored in the Healthcare PTN node can be encrypted as outlined in Section 4.3.3 such that no malicious entity can read the log records. Also, assuming the membership rules govern that any node that joins the PTN must be a hospital or some kind of healthcare provider, the gateway node can ensure that a node will not be able to join the Healthcare Provider's PTN unless it provides some sort of a certification that it is a healthcare provider.

Scientific Data Providers

Research data collection and usage is essential for the advancement of science. In many research experiments, different organizations collaborate in the collection, use and dissemination of research data. Science Commons has laid out the foundations for sharing and communicating the appropriate use of data to the researchers [125], and there has been huge successes in sharing genomics data using Science Commons [126].

However, there are currently no mechanisms for research institutions to track how their data was used in other institutions once they have shared it. The PTNs provide the mechanism in which the organizations can verify how their data was re-used, what

research results came out of the data they contributed to it, etc.

Additionally, each research institution will be able to communicate specific usage restrictions, which could be either set by the institution or by the participants of the study. This will limit any legal encumbrances associated with the data to the recipient research institutions as explicit permission has been granted and recorded in the PTN. Similarly, each research institution can give a transparent view of the data usages to their own participants. Studies on sexual behaviors, brain activity, psychological conditions, serious illnesses such as cancer, may collect very sensitive data from participants. By enabling the participants to set their own usage restrictions, and assuring them that they can audit their data at any time will empower them.

Social Media Websites

In the social networking sphere, users like to post things from news articles to photos and videos on their favorite social networking site to share with their friends and strangers alike. Therefore these users will appreciate a system that enables them to see how the content they have shared has been transferred across different social media websites. Similarly, for creative content providers, there are lot of incentives to determine the virality of content on the Web through social media. Their motivation for this could be rewards based, or it could simply be motivated by their inquisitiveness. If social media websites collaborate on a consortium to run a PTN, users will be able to determine what happened to their content after sharing. This architecture will sustain even in a PTN that has nodes from competing social networking services, where nodes can leave arbitrarily whenever they wish to, due to the inherent DHT architecture that is equipped to handle churn (the log records will be replicated, and other voluntary nodes can join and take part in the PTN).

7.2.2 Incentive Mechanisms

Utilizing technology to enable users to do the right thing empowers them instead of alienating them, and is more likely to promote openness and sharing leading to more creativity and economic value added. HTTPPA brings with it some desirable features that the current Web lacks. Targeting to application domains on how these features can be addressed is another strategy for adoption. Some of these features are described below.

Determine Trust and Data Quality

By knowing data provenance, we can judge the quality of the data and measure its trustworthiness. For example, if in an AIDS bulletin board all the posted information was stripped of its professional sources, users would be unable to distinguish whether the source of some 'cutting-edge' treatment is a layperson or an AIDS specialist. Thus, source cues and any other derivation information about the information resource can increase user confidence in the source and quality of information. HTTPPA based clients can retrieve these source information, and display them where necessary, thus increasing trust in the users.

Owners Can Determine Usages of Data

One of the primary goals in the design of the PTN is to use provenance information in determining the usages of data, i.e. to be able to infer who is responsible for the access, use, and modification of data, how and where it happened. Such an application can be used in addressing complex issues arising from data reuse such as privacy violations from the disclosure and transmission of sensitive data, as well as intellectual property rights violations on more progressive licensing schemes such as Creative Commons where there is no enforcement as was shown in this thesis.

Vendor-consumer relationship management in e-commerce using HTTPPA is one of the attractive directions to get the technology off the ground. Technologies such as *Do-Not-Track* [127] has not demonstrated its effectiveness [128]. But the questions as

to should organizations be allowed to put cookies on users' machines? should they be allowed to track the mouse movements of individual users visiting websites? should they be allowed to generate personal buying patterns of their online customers, then offer them individualized promotional offers? should they be allowed to share consumer information with other organizations? are still left unanswered. One solution these e-commerce organizations can provide is to give the users a complete view of how their personal information, or any data gleaned from them through the interactions on the website using HTTPPA.

Reduction of Data Handling Responsibility

There are many added benefits if HTTPPA were to be used in creating an Accountable System to manage the data collection and handling in research studies that collect lot of sensitive information. The client-side accountable tool run by the study participant will respond to the accountable server set up by the researchers for any data requests. The client can set usage restrictions on the data sent, that will better reflect what the study participants intend their data to be used for.

Also, often times participants in such studies get some form of compensation from the institutions that use their data for research purposes. The view of how their data was used could provide another compensation mechanism that is easily handled by HTTPPA. But more importantly, they, i.e. the participants, will be in control of their own data thus reducing any liability and the data handling responsibility for the researchers.

Monetary Incentives

There are many challenges in terms of the adoptability of this technology by major websites, and encouraging users to specify their usage restrictions up front. One way to tackle this would be to incorporate a payment mechanism to reward the distributors of data items that honor the protocol.

When we consider independent web content creators (e.g. bloggers, photographers, or artists) or large companies that create and distribute web content (Insta-

gram, Vine, Facebook), most current web content creators are supported through embedded third party ads, which tend to detract value from the content itself. When considering alternatives to ads, it is important to remember that all content on the web has overhead cost, from content creation time and effort, to server-side development and hosting, etc. So mechanisms to connect consumers with their preferred creators for a fair exchange benefits both the parties. There are popular methods for giving financial incentives directly to the source author such as Flattr [129] and GitTip [130]. Topsy, a project that I collaborated on, also provides a rewarding mechanism, but, instead of micro-payments, the focus was to give out lump sum payments determined through usage data reported using the client-side tools [131]. The Topsy client aggregates the accesses to a web page that has some machine readable code in RDFa, and sends out a payment at the end of a certain period. However, if we utilize HTTPa, the PTN can provide a more accurate and authoritative information about the usage of a resource compared to reports from web clients.

7.2.3 Policy Changes

Another important aspect for deployment is regulation and policy changes in governments and other authoritative organizations. As an example, the EU recently enacted a law for its citizens to be able to erase records on the Web [132] by asking search engines to remove them from their indices. Similarly, if there is enough demand from consumers that they wish to see how their data is used, governments can require organizations to utilize the PTN to provide auditing of information collected by them from the users.

In specific domains such as healthcare, there seem to be no apparent incentive for healthcare providers to adopt this architecture. However, at least in the US, there is a slow but steady push towards opening up personal health records to patients via the 'Blue Button Initiative' [133]. The data from blue-button enabled sites are meant to increase interaction among healthcare providers and other trusted entities. Since the data will no longer be locked up in silos, we can imagine a decentralized healthcare data eco-system evolving that entails complex usage scenarios. In such a

system, which may come to fruition in the near future, Accountable Systems can be readily adopted.

Ideally, the PTNs can be implemented by similar organizations that routinely share and access each other's data. This may be within organizations, or even among different governments. As new organizations start sharing data with each other, the PTNs too can merge, thus paving the way for the original vision of the global PTN.

7.3 Current Limitations

This section identifies some of the open issues in the work presented in this thesis. Some of these could be potential future work to further this work.

7.3.1 Scalability

Scalability can affect the growth of the PTN in three ways: limitations on the disk space allocation in the PTN, request latency to cater to high volumes, and handling of the potentially enormous log records. Each of these scalability issues are described in detail below.

Disk Space on the PTN Node

When deploying the PTN at web scale, there is a potential for the log records for a popular resource to grow exponentially. Thus the PTN node may run out of disk space at a certain time and the PTN node may fail. Failure of a single node will not hinder the operation of the PTN as a whole, since other nodes have already replicated the data stored on this failed node. Also, the hard disk prices are constantly going down, that we may assume unlimited storage on many computer systems that implement a PTN node. Maintaining the PTN should not be burdensome, and the PTN node maintainer should not be required to take care of failed disks too often.

Therefore, we suggest using graph summarization algorithms to suppress triples that were generated before a certain epoch and create a summary, while leaving the newer additions in triple format to answer any audit requests. Older entries can be

removed from the data store, and the summaries can be kept to answer any *audit* requests. But, auditing is inherently a historic function, where access to older records may be needed to give a complete picture for the usages of a resource. Therefore, access to summaries of graphs may not be sufficient in some cases.

Request Latency

As part of this thesis, I have done a scalability test on a PTN comprising of 100 nodes on PlanetLab, where the latency of requests to update and retrieve usage logs based on a workload over the course of 24 hours were continuously assessed. As the size of the log grows, the time remained almost comparable for our workload with only a slight overhead for HTTP *get* and *put* requests. However, I have not yet done a stress test of the PTN under very large number of requests on very large scale of usage log data. Also, if a node hosts both a web server and the PTN code, there may be a hit on the HTTP transactions, and/or the requests to the PTN, which is also not yet ascertained.

Handling Large Audit Log Records

Accountability log records for resources that have a high out-degree, such as a popular news items that get shared on social media over and over again, may have several thousand entries of uninteresting accesses. The owner may be interested in the aggregate summary of the logs rather than the actual log records. In such cases, a summarization algorithm can condense the results and present to the user only the interesting things. This is not currently handled, and the user/client tool may be overwhelmed in cases where there is a massive audit log for a single resource.

7.3.2 Security and Privacy Concerns

The PTN was initially designed to be deployed at global scale. Since the PTN records have the same structure, i.e. the value in the PTN record uses the Prov-O recommendation in defining the logs, many applications can share the PTN. For example,

Transparent Health, and the Accountable Meme Generator can both share the same PTN since the underlying log data structure and the interface to the PTN is similar, even though the corresponding applications are vastly different. Thus an interesting question arises as to the security and privacy of the audit logs. Clearly, Transparent Health demands more security and privacy of the log records compared to the Accountable Meme Generator.

When using HTTPPA to access and modify records in the PTN, the PTN needs the requestor (either the client or the server) to authenticate before performing the request. Therefore protocol participants will not be able to read each other's audit logs, and they will be only given their own audit logs after authentication. However, the current implementation of the PTN does not provide a complete solution to prevent a (malicious) super user in a PTN node from accessing the raw log data from disk.

The logs stored in the PTN are not as sensitive as the actual data items, but they can still pose serious security problems. With unencrypted log records a malicious node can read the disk, and find out the information about the fact that the subject of the log record had a visit to the oncologist, for example. Although the details of this visit is not available, as the actual medical record is available at the hospital, the mere record of the visit in the PTN can signal that the subject of this log record is a cancer patient.

In Transparent Health, the Accountable System was designed such that the patient (user) will perform requests to the PTN through the server, and there was no direct involvement of the client (i.e. the user-agent) with the PTN. Therefore, the records are encrypted using the server's keys as described in Section 4.3.3. In future systems, encrypting patient log data with the patient's public key could provide greater security. Some PTNs, for example the PTN used for the Accountable Meme Generator is unencrypted, as many of the requests are made by the client. Therefore, there is the possibility of a malicious node that have joined the PTN to read the records stored on its disk. This leads to a serious security and privacy concern for the PTN, that must be addressed in the next iteration of HTTPPA.

7.3.3 Liability Concerns

Some nodes in the PTN may not want to bear the responsibility of storing the meta-data about a potentially private information item. In such a case, the node may want to reject or probably not even be notified of that record. Even though this is not currently handled in the implementation given in this thesis, the PTN can be programmed such that the gateway nodes do not route these records to the corresponding PTN nodes that do not want any sensitive records stored in them.

7.3.4 Potential for PTN Abuse

Abusive use of the systems are possible if someone decides to send massive number of requests for explanations through the 'Audit' functionality, resulting in denial of service attacks on the PTN. These kinds of attacks have plagued the track-back and ping-back systems in the past [134]. However, unlike in those systems, the identity of the agents in the transparent system are known. Therefore, if someone issues too many requests, that client maybe banned from the system. This feature is not currently implemented and it is a possible future enhancement of the project.

7.3.5 Identity Requirements

High degrees of anonymity provides significant barriers to accountability in contexts involving software [135]. After all, if you do not know the identity of the other individuals reusing information in an accountable web-ecosystem, there are fewer incentives for individuals to behave with goodwill. The PTN architecture encompasses a technical solution to a social problem. In the current implementation, the identity of the individuals are required. One of the main challenges we foresee is the reluctance for someone to behave freely knowing that accesses and usages of their records are logged. Therefore, Accountable Systems may not be suitable for the Web in general, but probably suited to specific sub domains such as the two problems outlined in this thesis where there is clear incentive to using the real identity of an individual.

In order to provide anonymous or pseudonymous identity support, Accountable

Systems can borrow the notion of “accountable anonymity” as used in [136] where a participant remains anonymous unless he breaks the rules and disrupts the system, at which point his identity is revealed.

7.4 Future Work

Several avenues of investigation arise from the work described in this thesis. As mentioned in Section 7.2, Accountable systems can be extended to many other applications such as healthcare, research data collection and usage, financial information exchanges for credit reporting, social networking, creative content networks, etc. Implementing PTNs specific to these application domains along with the corresponding client and server applications that utilizes HTTPPA will lead to the adoption of this technology. This will bring many benefits to both the consumers and the producers of web content. In addition, as mentioned in Section 7.3, the current implementation of Accountable Systems has some limitations with respect to scalability, security, liability, possible resource abuse and identity management. While some of these limitations can be solved technically, some others will need some social ingenuity. We leave these for a future iteration of HTTPPA.

7.5 Conclusion

HTTPPA can potentially enable market and regulatory forces to identify and warn or punish users who misuse information on the Web. I believe that healthcare providers, government organizations, academic institutions, and businesses will be the early adopters of this accountable web protocol with usage restriction management within their networks that are served by their own PTN. On the longer run, in a similar vein in which the growth of e-commerce web sites led to the massive adoption of HTTPS, I envision that HTTPPA will be accepted by the larger web community at a time when without it, the web’s growth would be hampered by privacy and intellectual property rights problems.

Appendix A

Summary of User Generated Content Websites Survey

Website	Licenses available in the website	Support for remixing	Mechanisms of credit giving
YouTube	Public Domain, Creative Commons (only for special partners)	Users have to download to remix.	Utilizes ContentID and lets partners choose to either block, monetize or advertise.
Vimeo	No licenses are allowed.	Not available	Manual (it lets people give credit to other Vimeo users.)
Flickr	Creative Commons	Not available	Not available
Photo Bucket	A 'free' license which allows the website, and it's users to reuse content.	Not available	Not available

DeviantArt	Creative Commons	Available via their premium “Deviations” interface.	Available via their premium “Deviations” interface.
CCMixer	Creative Commons	Available (does not provide an interface to remix, the components of the remix can be attributed to other works.)	Manual (It lets people give credit to other CCMixer users.)
Indaba Music	Creative Commons (Attribution and Non-commercial use only)	Not available	Manual
Twitter	None.	Available (using ‘RT’ or ‘via’)	Automatic Retweet gives automatic credit. Manual retweeters use the RT @originator convention.
Identi.ca	Creative Commons	Available (same as twitter, slightly different terminology)	Can be rebroadcast to the user’s followers (same as twitter, slightly different terminology.)
My Experiment	CC, MIT, BSD, GPL, LGPL, Apache, PS	Available	Available

Table A.1: User Generated Content Websites Survey

Appendix B

License Lookup Table

		<i>Component License 1</i>									
	PD	CC0	BY	BY-NC	BY-NC-ND	BY-NC-ND-SA	BY-NC-SA	BY-ND	BY-ND-SA	BY-SA	ARR
PD	PD	CC0	BY	BY-NC	x	x	BY-NC-SA	x	x	BY-SA	ARR
CC0	CC0	CC0	BY	BY-NC	x	x	BY-NC-ND-SA	x	x	BY-SA	ARR
BY	BY	BY	BY	BY-NC	x	x	BY-NC-ND-SA	x	x	BY-SA	x
BY-NC	BY-NC	BY-NC	BY-NC	BY-NC	x	x	BY-NC-SA	x	x	x	x
BY-NC-ND	x	x	x	x	x	x	x	x	x	x	x
BY-NC-ND-SA	x	x	x	x	x	x	x	x	x	x	x
BY-NC-SA	BY-NC-SA	BY-NC-SA	BY-NC-SA	BY-NC-SA	x	x	BY-NC-SA	x	x	x	x
BY-ND	x	x	x	x	x	x	x	x	x	x	x
BY-ND-SA	x	x	x	x	x	x	x	x	x	x	x
BY-SA	BY-SA	BY-SA	BY-SA	x	x	x	x	x	x	BY-SA	x
ARR	ARR	ARR	x	x	x	x	x	x	x	x	ARR

Legend:
PD: Public Domain
CC0: Creative Common
BY: Attribution
NC: Non Commercial
ND: No Derivatives
SA: Share Alike
ARR: All Rights Reserve
x: Cannot be combin

Table B.1: License Lookup Table

Bibliography

- [1] Berners-Lee, Timothy J. Information Management: A proposal – oai:cds.cern.ch:369245. Technical Report CERN-DD-89-001-OC, CERN, Geneva, Mar 1989.
- [2] danah m. boyd and Nicole B. Ellison. Social Network Sites: Definition, History, and Scholarship. *Journal of Computer-Mediated Communication*, 13(1):210–230, 2007.
- [3] Catherine Dwyer, Starr Hiltz, and Katia Passerini. Trust and Privacy Concern Within Social Networking Sites: A Comparison of Facebook and MySpace. In *Proceedings of the Thirteenth Americas Conference on Information Systems, Keystone, Colorado*, 2007.
- [4] Prema Nakra. Consumer privacy rights: CPR and the age of the Internet. *Management Decision*, 39(4):272–279, 2001.
- [5] Federal Trade Commission. Privacy online: Fair information practices in the electronic marketplace: A federal trade commission report to congress. <http://www.ftc.gov/reports/privacy-online-fair-information-practices-electronic-marketplace-federal-trade-commission>.
- [6] George J Annas. Hipaa regulations-a new era of medical-record privacy? *New England Journal of Medicine*, 348(15):1486–1490, 2003.
- [7] Ajit Appari and M Eric Johnson. Information security and privacy in health-care: current state of research. *International journal of Internet and enterprise management*, 6(4):279–314, 2010.
- [8] Erin McCann. Healthcare IT News. <http://www.healthcareitnews.com/news/four-year-long-hipaa-data-breach-discovered>, August 2014.
- [9] Kasey Jones. Dr. nikita levy, Johns Hopkins gynecologist, allegedly hid camera inside pen to secretly tape patients. <http://www.huffingtonpost.com/2013/02/27/dr-nikita-levy-johns-hopk-n-2776805.html>, February 2013.
- [10] CBS News. Maria Shriver’s medical records leaked. <http://www.cbsnews.com/news/maria-shrivers-medical-records-leaked/>, April 2008.

- [11] Reuters. Hospital workers fired over privacy breach reportedly targeted at kim kardashian. <http://www.huffingtonpost.com/2013/07/14/workers-fired-kim-kardashian-n-3592841.html>, July 2013.
- [12] CNN. 27 suspended for Clooney file peek. <http://www.cnn.com/2007/SHOWBIZ/10/10/clooney.records/index.html?eref=ew>, October 2007.
- [13] Charles Ornstein. Ex-worker indicted in celebrity patient leaks. <http://articles.latimes.com/2008/apr/30/local/me-ucla30>, April 2008.
- [14] Sam Narisi. Hospital employees fired for snooping. <http://www.healthcarebusinessstech.com/hospital-employees-fired-for-snooping/>, November 2012.
- [15] Lev Manovich. Remixability and remixability, http://www.manovich.net/DOCS/Remix_modular.doc. 2005.
- [16] Yochai Benkler. *The Wealth of Networks: How Social Production Transforms Markets and Freedom*. Yale University Press, 2006.
- [17] Lawrence Lessig. *Remix: Making Art and Commerce Thrive in the Hybrid Economy*. Penguin Press HC, The, Oct 2008.
- [18] Pamela Samuelson. Preliminary thoughts on copyright reform. *Utah L. Rev.*, page 551, 2007.
- [19] Sal Humphreys. The challenges of intellectual property for users of social networking sites: a case study of ravelry. In *Proceedings of the 12th international conference on Entertainment and media in the ubiquitous era*, pages 125–130. ACM, 2008.
- [20] Jenna Wortham and Nick Bolton. What instagams new terms of service mean for you. <http://bits.blogs.nytimes.com/2012/12/17/what-instagams-new-terms-of-service-mean-for-you/>. Accessed: 06-17-2014.
- [21] Jerome H Saltzer and Michael D Schroeder. The protection of information in computer systems. *Proceedings of the IEEE*, 63(9):1278–1308, 1975.
- [22] Joseph Y Halpern and Judea Pearl. Causes and explanations: A structural-model approach. part i: Causes. *The British journal for the philosophy of science*, 56(4):843–887, 2005.
- [23] Tomasz Truderung, Andreas Vogt, et al. Accountability: definition and relationship to verifiability. In *Proceedings of the 17th ACM conference on Computer and communications security*, pages 526–535. ACM, 2010.

- [24] Deepak Garg, Limin Jia, and Anupam Datta. Policy auditing over incomplete logs: theory, implementation and applications. In *Proceedings of the 18th ACM conference on Computer and communications security*, pages 151–162. ACM, 2011.
- [25] Helen Nissenbaum. *Privacy in context: Technology, policy, and the integrity of social life*. Stanford University Press, 2009.
- [26] Daniel J. Weitzner, Harold Abelson, Tim Berners-Lee, Joan Feigenbaum, James Hendler, and Gerald Jay Sussman. Information accountability. *Commun. ACM*, 51(6):82–87, June 2008.
- [27] Joan Feigenbaum, James A Hendler, Aaron D Jaggard, Daniel J Weitzner, and Rebecca N Wright. Accountability and deterrence in online life. In *Proceedings of the 3rd International Conference on Web Science, ACM*, 2011.
- [28] Joan Feigenbaum, Aaron D Jaggard, and Rebecca N Wright. Towards a formal model of accountability. In *Proceedings of the 2011 workshop on New security paradigms workshop*, pages 45–56. ACM, 2011.
- [29] Joan Feigenbaum, Aaron D Jaggard, Rebecca N Wright, and Hongda Xiao. Systematizing accountability in computer science (version of feb. 17, 2012). Technical report, Citeseer, 2012.
- [30] Adam Barth, John C Mitchell, Anupam Datta, and Sharada Sundaram. Privacy and utility in business processes. *CSF*, 7:279–294, 2007.
- [31] Michael Backes, Anupam Datta, Ante Derek, John C Mitchell, and Mathieu Turuani. Compositional analysis of contract signing protocols. In *Computer Security Foundations, 2005. CSFW-18 2005. 18th IEEE Workshop*, pages 94–110. IEEE, 2005.
- [32] Andreas Haeberlen, Petr Kouznetsov, and Peter Druschel. Peerreview: Practical accountability for distributed systems. In *ACM SIGOPS Operating Systems Review*, volume 41, pages 175–188. ACM, 2007.
- [33] Jaehong Park and Ravi Sandhu. The ucon abc usage control model. *ACM Transactions on Information and System Security (TISSEC)*, 7(1):128–174, 2004.
- [34] Alexander Pretschner, Manuel Hilty, and David Basin. Distributed usage control. *Communications of the ACM*, 49(9):39–44, 2006.
- [35] P. Kumari, A. Pretschner, J. Peschla, , and J.-M. Kuhn. Distributed data usage control for web applications: a social network implementation. In *Proceedings of the First ACM Conference on Data and Application Security and Privacy*, pages 85–96, 2011.

- [36] Roy Fielding, Jim Gettys, Jeffrey Mogul, Henrik Frystyk, Larry Masinter, Paul Leach, and Tim Berners-Lee. Hypertext transfer protocol–http/1.1, 1999.
- [37] Oshani Seneviratne and Andres Monroy-Hernandez. Remix culture on the web: A survey of content reuse on different User-Generated content websites. In *Web Science Conference at World Wide Web Conference 2010*, April 2010.
- [38] Benjamin Mako Hill, Andrés Monroy-Hernández, and Kristina Olson. Responses to remixing on a social media sharing website. In *ICWSM*, 2010.
- [39] Oshani Seneviratne, Lalana Kagal, and Tim Berners-Lee. Policy-Aware Content Reuse on the Web. In *ISWC 2009*, pages 553–568, 2009.
- [40] Piero A. Bonatti, Claudiu Duma, Norbert E. Fuchs, Wolfgang Nejdl, Daniel Olmedilla, Joachim Peer, and Nahid Shahmehri. Semantic web policies - a discussion of requirements and research issues. In *ESWC*, pages 712–724, 2006.
- [41] Mitchell D Swanson, Mei Kobayashi, and Ahmed H Tewfik. Multimedia data-embedding and watermarking technologies. *Proceedings of the IEEE*, 86(6):1064–1087, 1998.
- [42] Neal R Wagner. Fingerprinting. In *2012 IEEE Symposium on Security and Privacy*, pages 18–18. IEEE Computer Society, 1983.
- [43] Puneet Kishor. Babel of licenses. <http://www.punkish.org/Babel-of-Licenses>. Accessed: 07-02-2014.
- [44] Hal Abelson, Ben Adida, Mike Linksvayer, Nathan Yergler. ccREL: The Creative Commons Rights Expression Language, <http://wiki.creativecommons.org/images/d/d6/Ccrel-1.0.pdf>. *Creative Commons Wiki*, 2008.
- [45] Creative Commons Customized Search in Google. <http://creativecommons.org/press-releases/entry/5692>.
- [46] Flickr API. <http://www.flickr.com/services/api>.
- [47] blip.tv - Hosting, distribution and advertising platform for creators of web shows. <http://blip.tv>.
- [48] OWL Music Search. <http://www.owlmusicsearch.com>.
- [49] SpinXpress - Collaborative media production platform. <http://spinexpress.com>.
- [50] RDFa, Resource Description Framework in Attributes. <http://www.w3.org/2006/07/SWD/RDFa/syntax>.
- [51] How to attribute Flickr images. <http://www.squidoo.com/cc-flickr#module12311035>.

- [52] Creative Commons BY 3.0 Unported Legal Code. <http://creativecommons.org/licenses/by/3.0/legalcode>.
- [53] Tim Berners-Lee and James Hollenbach and Kanghao Lu and Joe Presbrey and Eric Prud'ommeaux and mc schraefel. Tabulator Redux: Browning and Writing Linked Data . In *Linked Data on the Web Workshop at WWW08*, 2008.
- [54] MozCC - Firefox extension to discover Creative Commons licenses. <http://wiki.creativecommons.org/MozCC>.
- [55] Think Free - Java based web office suite. <http://www.thinkfree.com>.
- [56] Flickr image reuse for openoffice.org. <http://wiki.creativecommons.org/Flickr-Image-Re-Use-for-OpenOffice.org>.
- [57] Attributor - Subscription based web monitoring platform for content reuse detection. <http://www.attributor.com>.
- [58] picScout - Image tracker for stock photography agencies and professional photographers. <http://www.picscout.com>.
- [59] Ken Doctor, Blog Entry on "Attributor Fair Syndication Consortium Completes Newspaper Trifecta". <http://www.contentbridges.com/2009/04/attributor-ad-push-on-piracy-completes-newspaper-trifecta.html>.
- [60] Harvey C Jones. XHTML documents with inline, policy-aware provenance. Master's thesis, Massachusetts Institute of Technology, May 2007.
- [61] Puneet Kishor, Oshani Seneviratne, and Noah Giansiracusa. Policy aware geospatial data. *arXiv preprint arXiv:1304.5755*, 2013.
- [62] Harlan Onsrud, Gilberto Camara, James Campbell, and Narnindi Sharad Chakravarthy. Public commons of geographic data: Research and development challenges. In *Geographic information science*, pages 223–238. Springer, 2004.
- [63] David McCurry, James Campbell, M Lutz, Harlan Onsrud, and Kenton Williams. Managing intellectual property issues in a commons of geographic data. In *Digital Libraries, 2006. JCDL'06. Proceedings of the 6th ACM/IEEE-CS Joint Conference on*, pages 347–347. IEEE, 2006.
- [64] Jerome H Reichman and Paul F Uhler. A contractually reconstructed research commons for scientific data in a highly protectionist intellectual property environment. *Law and Contemporary Problems*, pages 315–462, 2003.
- [65] Jeff de la Beaujardiere. OGC web map service interface. <http://www.opengeospatial.org/standards/wms>, 2004.
- [66] Inc Environmental Systems Research Institute. ESRI shapefile technical description. <http://www.esri.com/library/whitepapers/pdfs/shapefile.pdf>, 1998.

- [67] Jessica Staddon, Philippe Golle, and Bryce Zimny. Web-based inference detection. *SS*, 7:1–16, 2007.
- [68] Arvind Narayanan and Vitaly Shmatikov. De-anonymizing social networks. In *Security and Privacy, 2009 30th IEEE Symposium on*, pages 173–187. IEEE, 2009.
- [69] Oshani Wasana Seneviratne. Augmenting the web with accountability. In *Proceedings of the 21st international conference companion on World Wide Web*, pages 185–190. ACM, 2012.
- [70] Roy Thomas Fielding. *Architectural styles and the design of network-based software architectures*. PhD thesis, University of California, Irvine, 2000.
- [71] B Lampson. Usable security: how to get it. *Communications of the ACM*, Jan 2009.
- [72] Oshani Seneviratne and Lalana Kagal. Addressing Data Reuse Issues at the Protocol Level. In *POLICY 2011, IEEE International Symposium on Policies for Distributed Systems and Networks*, pages 141–144, 2011.
- [73] Ted Kang and Lalana Kagal. Enabling Privacy-awareness in Social Networks. In *Intelligent Information Privacy Management Symposium at the AAAI Spring Symposium 2010*, March 2010.
- [74] Sebastian Tramp, Henry Story, Andrei Sambra, Philipp Frischmuth, Michael Martin, Sören Auer, et al. Extending the webid protocol with access delegation. In *Proceedings of the Third International Workshop on Consuming Linked Data (COLID2012)*, 2012.
- [75] Dick Hardt. The OAuth 2.0 authorization framework. <http://oauth.net/2>, 2012.
- [76] Sean Rhea, Brighten Godfrey, Brad Karp, John Kubiawicz, Sylvia Ratnasamy, Scott Shenker, Ion Stoica, and Harlan Yu. Opendht: a public dht service and its uses. In *ACM SIGCOMM Computer Communication Review*, volume 35, pages 73–84. ACM, 2005.
- [77] SEDA) Sean Rhea (project based on OceanStore. The bamboo distributed hash table, a robust, open-source dht. <http://bamboo-dht.org/>. Accessed: September 2011.
- [78] Timothy Lebo, Satya Sahoo, and Deborah McGuinness. PROV-O: The provenance ontology. <http://www.w3.org/TR/prov-o/>, 2003.
- [79] Brent Chun, David Culler, Timothy Roscoe, Andy Bavier, Larry Peterson, Mike Wawrzoniak, and Mic Bowman. Planetlab: an overlay testbed for broad-coverage services. *SIGCOMM Comput. Commun. Rev.*, 33:3–12, July 2003.

- [80] Tim Berners-Lee. The “oh, yeah?” -button. <http://www.w3.org/DesignIssues/UI.html#OhYeah>, 1997.
- [81] Oshani Seneviratne and Lalana Kagal. Enabling privacy through transparency. In *Privacy Security and Trust*, pages (in–press), 2014.
- [82] Patrick Kierkegaard. Electronic health record: Wiring europes healthcare. *Computer Law & Security Review*, 27(5):503–515, 2011.
- [83] Ashish K Jha, Timothy G Ferris, Karen Donelan, Catherine DesRoches, Alexandra Shields, Sara Rosenbaum, and David Blumenthal. How common are electronic health records in the united states? a summary of the evidence. *Health Affairs*, 25(6):w496–w507, 2006.
- [84] Lorrie Faith Cranor. Web privacy with Platform for Privacy Preferences. *Oreilly Books*, Jan 2002.
- [85] Find web sites that respect your privacy) - <http://www.privacybird.org/>.
- [86] Electronic Privacy Information Center. Pretty Poor Privacy: An Assessment of P3P and Internet Privacy. June 2000.
- [87] Phil Archer, Kevin Smith, and Andrea Perego. Protocol for Web Description Resources (POWDER): Description Resources. <http://www.w3.org/TR/powder-dr>.
- [88] Aza Raskin and Arun Ranganathan. Privacy: A Pictographic Approach. *W3C Workshop on Privacy for Advanced Web APIs*, 2010.
- [89] Primelife. D. Dashboard. <http://www.primelife.eu/results/opensource/76-dashboards>.
- [90] Jorge R. Cuellar, John B. Morris, Deirdre K. Mulligan, Jon Peterson, and James M. Polk. Geopriv Requirements. Internet RFC 3693.
- [91] Andrei Popescu. Geolocation API Specification.
- [92] Nick Doty and Erik Wilde. Geolocation privacy and application platforms. In *Proceedings of the 3rd ACM SIGSPATIAL International Workshop on Security and Privacy in GIS and LBS*, SPRINGL ’10, pages 65–69, New York, NY, USA, 2010. ACM.
- [93] E Wilde. Simple policy negotiation for location disclosure. *w3.org*.
- [94] Tobias Pulls. Privacy-preserving transparency-enhancing tools. 2012.
- [95] Stefan Sackmann, Jens Strüker, and Rafael Accorsi. Personalization in privacy-aware highly dynamic systems. *Communications of the ACM*, 49(9):32–38, 2006.

- [96] Wolfgang Nejdl, Daniel Olmedilla, and Marianne Winslett. Peertrust: Automated trust negotiation for peers on the semantic web. In *Secure Data Management*, pages 118–132. Springer, 2004.
- [97] Pradeep Ray and Jaminda Wimalasiri. The need for technical solutions for maintaining the privacy of ehr. In *Engineering in Medicine and Biology Society, 2006. EMBS'06. 28th Annual International Conference of the IEEE*, pages 4686–4689. IEEE, 2006.
- [98] Corey M Angst and Ritu Agarwal. Adoption of electronic health records in the presence of privacy concerns: the elaboration likelihood model and individual persuasion. *MIS quarterly*, 33(2):339–370, 2009.
- [99] Shivaramakrishnan Narayan, Martin Gagné, and Reihaneh Safavi-Naini. Privacy preserving ehr system using attribute-based infrastructure. In *Proceedings of the 2010 ACM workshop on Cloud computing security workshop*, pages 47–52. ACM, 2010.
- [100] Helma van der Linden, Dipak Kalra, Arie Hasman, and Jan Talmon. Inter-organizational future proof ehr systems: a review of the security and privacy related issues. *International journal of medical informatics*, 78(3):141–160, 2009.
- [101] Randike Gajanayake, Renato Iannella, William B. Lane, and Tony R. Sahama. Accountable-ehealth systems : the next step forward for privacy. In *1st Australian eHealth Informatics and Security Conference*, Novotel Perth Langley, Perth, WA, November 2012. Edith Cowan University. 1st Australian eHealth Informatics and Security Conference is one of the six hosted conferences in secau Security Congress 2012.
- [102] Qiong Liu, Reihaneh Safavi-Naini, and Nicholas Paul Sheppard. Digital rights management for content distribution. In *Proceedings of the Australasian information security workshop conference on ACSW frontiers 2003-Volume 21*, pages 49–58. Australian Computer Society, Inc., 2003.
- [103] Commons Machinery. Building the Infrastructure of the Commons. <http://commonsmachinery.se/labs/>.
- [104] Susan Landau. Support for Fair Use with Project DReaM. *Sun Microsystems Laboratories*, <http://www.docstoc.com/docs/73060519/Support-for-Fair-Use-with-Project-DReaM>, Version 1.0 Rev A, April 2008.
- [105] Jong Wook Kim, K. Seluk Candan, and Junichi Tatemura. Efficient overlap and content reuse detection in blogs and online news articles. In *18th International World Wide Web Conference (WWW2009)*, April 2009.
- [106] N. Shivakumar and H. Garcia-Molina. Scam: A copy detection mechanism for digital documents. In *Second Annual Conference on the Theory and Practice of Digital Libraries*, 1995.

- [107] Yogesh L Simmhan, Beth Plale, and Dennis Gannon. A survey of data provenance in e-science. *ACM Sigmod Record*, 34(3):31–36, 2005.
- [108] Juliana Freire, David Koop, Emanuele Santos, and Cláudio T Silva. Provenance for computational tasks: A survey. *Computing in Science & Engineering*, 10(3):11–21, 2008.
- [109] Susan B Davidson and Juliana Freire. Provenance and scientific workflows: challenges and opportunities. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1345–1350. ACM, 2008.
- [110] Stuart Haber and W Scott Stornetta. *How to time-stamp a digital document*. Springer, 1991.
- [111] Luc Moreau, Paul Groth, Simon Miles, Javier Vazquez-Salceda, John Ibbotson, Sheng Jiang, Steve Munroe, Omer Rana, Andreas Schreiber, Victor Tan, et al. The provenance of electronic data. *Communications of the ACM*, 51(4):52–58, 2008.
- [112] Paul Groth, Yolanda Gil, James Cheney, and Simon Miles. Requirements for provenance on the web. *International Journal of Digital Curation*, 7(1):39–56, 2012.
- [113] Sara Magliacane. Reconstructing provenance. In *The Semantic Web–ISWC 2012*, pages 399–406. Springer, 2012.
- [114] Paul Groth. Provenancejs: Revealing the provenance of web pages. In *Provenance and Annotation of Data and Processes*, pages 283–285. Springer, 2010.
- [115] Brian Tierney, William Johnston, Brian Crowley, Gary Hoo, Chris Brooks, and Dan Gunter. The netlogger methodology for high performance distributed systems performance analysis. In *High Performance Distributed Computing, 1998. Proceedings. The Seventh International Symposium on*, pages 260–267. IEEE, 1998.
- [116] Paul Groth, Simon Miles, and Luc Moreau. Preserv: Provenance recording for services. 2005.
- [117] Reza Samavi and Mariano P Consens. L2tap+ scip: An audit-based privacy framework leveraging linked data. In *Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom), 2012 8th International Conference on*, pages 719–726. IEEE, 2012.
- [118] Deborah L McGuinness and Paulo Pinheiro da Silva. Explaining answers from the semantic web: The inference web approach. *Web Semantics: Science, Services and Agents on the World Wide Web*, 1(4):397–413, 2004.

- [119] Paulo Pinheiro Da Silva, Deborah L McGuinness, and Richard Fikes. A proof markup language for semantic web services. *Information Systems*, 31(4):381–395, 2006.
- [120] XMP - Extensible Metadata Platform. <http://www.adobe.com/products/xmp/index.html>.
- [121] Exchangeable Image File Format . <http://www.exif.org/specifications.html>.
- [122] International Press Telecommunications Council Photo Metadata Format. <http://www.iptc.org/IPTC4XMP>.
- [123] Andy Bavier. Planetlab. <http://www.planet-lab.org/consortium>. Accessed: 08-24-2014.
- [124] Health data consortium. <http://www.healthdataconsortium.org/>. Accessed: 08-24-2014.
- [125] John Wilbanks and James Boyle. Introduction to science commons. *online under http://sciencecommons.org/wp-content/uploads/ScienceCommons_Concept_Paper.pdf*, retrieved, 19:2010, 2006.
- [126] Robert Cook-Deegan. The science commons in health research: structure, function, and value. *The Journal of Technology Transfer*, 32(3):133–156, 2007.
- [127] The History of the Do Not Track Header. <http://paranoia.dubfire.net/2011/01/history-of-do-not-track-header.html>.
- [128] Zach Miners. Internet 'Do Not Track' system is in shatters. http://www.computerworld.com/s/article/9248503/Internet_Do_Not_Track_system_is_in_shatters.
- [129] Flatlr - support creators directly on services you like. <https://flatlr.com>.
- [130] Gittip - weekly payments motivated by gratitude. <https://www.gittip.com/>.
- [131] Doc Searls. Emancipay: A Relationship Management and Voluntary Payment Framework. *Harvard Law Blog*, 2010.
- [132] DAVID STREITFELD. European court lets users erase records on web. <http://www.nytimes.com/2014/05/14/technology/google-should-erase-web-links-to-some-personal-data-europes-highest-court-says.html>. Accessed: 08-25-2014.
- [133] Blue button. <http://www.whitehouse.gov/open/innovations/BlueButton>, March 2014.

- [134] Elie Bursztein, Peifung E Lam, and John C Mitchell. Trackback spam: Abuse and prevention. In *Proceedings of the 2009 ACM workshop on Cloud computing security*, pages 3–10. ACM, 2009.
- [135] Helen Nissenbaum. Accountability in a computerized society. *Science and engineering ethics*, 2(1):25–42, 1996.
- [136] Henry Corrigan-Gibbs and Bryan Ford. Dissent: accountable anonymous group messaging. In *Proceedings of the 17th ACM conference on Computer and communications security*, pages 340–350. ACM, 2010.