Automated Tracking of Online Service Policies

J. Trent Adams
The Internet Society
adams@isoc.org

Kevin Bauer
University of Colorado
bauerk@colorado.edu

Asa Hardcastle
OpenLiberty.org
asa.openliberty@zenn.net

Dirk Grunwald
University of Colorado
grunwald@colorado.edu

Douglas Sicker
University of Colorado
sicker@colorado.edu

ABSTRACT

The use of online services such as social networking sites or search engines is generally governed by one or more usage policies, such as terms of service, end-user license agreements, and privacy policies. To facilitate effective dissemination of policy information, structured policy data formats such as the Platform for Privacy Preferences (P3P) and others have been proposed, but have not benefited from wide adoption. Consequently, today users still must resort to locating, reading, and understanding the usage policies for the services they use regularly—an onerous task for ordinary users.

While structured policy formats have been slow to be adopted, human-readable natural language policy documents are ubiquitous across a wide variety of online services. For example, these documents can express a service’s privacy policy, end-user license agreement, or terms of service. In this paper, we present the design and implementation of the Policy Audit System, a system that aggregates the natural language policies for a wide variety of online services, monitors these policies for changes over time, and informs interested users of salient policy changes in a manner that is easy to understand.

1. INTRODUCTION

Today Internet users interact with a wide variety of online services on a regular basis including social networking sites, collaboration tools, banking, and search engines. The use of these services is typically governed by one or more usage policies, such as terms of service, end-user license agreements, and privacy policies. Some of these policies are mandated by regulation, while others are developed by the service provider themselves to clarify how users are allowed to interact with their service. When users visit these sites, they implicitly—and often unknowingly—accept the policies.

Online services that publish usage policies expect them to be read, understood, and accepted for users to continue using their service. However, research shows that even those users who read policies may not fully understand their implications [33], and the task of reading policies has been argued to result in significant losses of time and money [32]. Many users remain unaware of the usage policies or their implications until there is a reported problem that threatens the user’s security, privacy, rights, or other expectations.

An obvious example of when users become aware of a service’s usage policies is when the service experiences a data breach. In this event, users may question the site’s published data retention policies in an effort to understand what personal data may be compromised. A less obvious example is when usage policies include language stating that the service reserves rights to use personal data in a way that is counter to the user’s expectations. Further examples include similar issues arising when a usage policy changes, even after the user reads, understands, and agrees to it.

In all of these cases, services and users share a common desire for the clear articulation of usage policies. Service providers need their users to understand the implications of using their service or they may face potential loss of revenue, increased costs to address problems, or even possible legal action. Similarly, users need to be effectively informed about the consequences (positive and negative) of using a particular service so they can make informed decisions when selecting services, prior to problems occurring.

Ideally, online services should publish their usage policies in common, well-known locations so they can be easily found. Also, their policies should be expressed using a standardized, structured ontology or data format. In this way, the policies could be automatically located, parsed, and semantically examined in a way that aids users awareness and understanding. The Platform for Privacy Preferences (P3P) is one such standardized format that allows online services to express their privacy policies in a manner that enables automatic processing and summarization of the policy to users [22]. With P3P, users can define a set of privacy preferences and choose to interact only with services that respect their preferences.

Unfortunately, while P3P is perhaps the most widely used structured policy schema on the web today, prior work has shown that it has achieved only limited adoption [27]. Furthermore, P3P as a structured policy format only expresses privacy information, not the kinds of general policy information that may be found in a terms of service policy or an end-user license agreement. A few examples of such information that users may wish to know, but are beyond the scope of the P3P schema, include billing and fee information or the conditions that may warrant the termination of service.

Given the benefits of a general-purpose structured policy format, many solutions have been proposed including components within the Kantara Initiative’s User-Managed Access [19] and Information Sharing [7] projects. However, these proposals are still in the early stages of design and development and, thus, there is currently no ubiquitous structured format for specifying general-purpose policy information.

We argue that until well-defined, structured approaches are widely adopted, service usage policies will continue to be
written in natural language and be published in an ad-hoc manner. While new approaches to improving the readability of service usage policies are in development, tools must be created to help users understand the policies currently available today.

To that end, we present the Policy Audit System to actively monitor changes in policy documents for a diverse set of popular online services. This system consists of three components. First, the Policy Monitor tracks changes in policy documents over time and records these changes in the Policy Library. Users can then interact with the tracked policies through a Policy Audit Client. We present one such client as a plugin running in the user’s web browser. In this case, when the user accesses an online service whose policy is tracked, the library is queried by the plugin. If a change in policy has occurred since the user’s previous visit, the changes are presented to the user as highlighted text, enabling interested users to identify and read the changes.

Further, we propose that the Policy Audit System can be used as a foundation upon which research analytics can be performed in an effort to explore how users engage with online service usage policies. The output of this research should benefit the design, development, and adoption of standardized mechanisms that help service providers more effectively communicate their usage policies with their users.

2. BACKGROUND

Online services often wish to express their policies describing how they are to interact with their users. These policies are contained within human-readable terms of service, end-user license agreement, and privacy policy documents. In the remainder of this paper, we generically refer to any one of these documents as a service’s usage policy.

2.1 The Need for Clear Policy Documents

During typical web browsing activities, users may unknowingly reveal a significant amount of identifying information to the websites they visit. Examples of such identifying information may include the users’ IP address and information exposed in the HTTP request headers often revealing their user-agent and version, operating system and version, local time and local language, and referring site. Using this information alone, websites can construct an identifying profile for users that visit their sites [11]. Additional identifiers can be embedded in tracking cookies, which simplify the task of identifying and profiling users’ behaviors.

Given the inherent privacy risks that result from ordinary web browsing, many websites explicitly state their policies with regard to their handling of personally identifiable information (PII) in a privacy policy document. Websites often provide a high-level summary of their policies with regard to topics such as what types of PII are collected, how long PII is retained, and how PII is shared.

However, to read a privacy policy the user must visit the site, find the privacy policy document, and comprehend the document’s contents. Furthermore, while the user is reading the privacy policy to decide whether they wish to use the website’s services, they may be revealing PII before they understand the site’s policies.

Within the P3P model, Privacy Finder [14] largely mitigates this problem. However, as we observe in Section 2.2, P3P policies themselves are not widely available.

2.2 Regulatory Environments

It is clear that site usage policies exist in large part due to lawyers advising service providers, basing their advice on government regulations and historical legal precedents. This advice is often informed by years of experience in how companies and individuals have interacted off-line. As society shifts more of its interactions on-line, companies are confronted with how to adapt their services to their users’ needs and expectations while remaining compliant with regulations that are also shifting at the same time.

Assuming that businesses will continue to expand online services, and that regulations will continue to govern their actions, usage policies will continue to exist. The goal in the development of usage policies is for both service providers and their users to effectively set the expectations of their engagement. As services become more interconnected, and data flows more easily between them and their users, it becomes necessary to develop mechanisms that minimize risk for all parties.

Key to this effort is a common syntax for describing policies, and mechanisms for them to be easily distributed. This style of solution becomes even more important within a global context where there are various regulatory environments involved. Solutions addressing effective usage policy communication need to be easy for a provider to support, clear for users to understand, and flexible when addressing various regulations.

2.3 Platform for Privacy Preferences

One effort to encode policy information and help users find, read, and understand website privacy policy documents is the platform for privacy preferences (P3P). It is an extensible Markup Language (XML) schema introduced by the World Wide Web Consortium (W3C) to automate and simplify the dissemination of policy information with regard to users’ privacy. The W3C standardized P3P 1.0 [21] in April 2002 and updated the standard with P3P 1.1 [22] in November 2006 to support various extensions.

P3P describes websites’ policies about the types of personal information that may be collected, how that information may be used, what rules regulate information sharing, how long personal data may be retained, and policies regarding an individual’s ability to access their own information. P3P XML documents are typically located at a well-known location within the website’s URL (e.g., /w3c/p3p.xml), enabling it to be easily found and automatically parsed. Alternatively, the URL of a website’s P3P document may be embedded in a page’s HTML source using the <link> tag.

In addition to the full P3P XML document, P3P elements may be specified as HTTP parameters. These compact P3P policies usually provide more limited policy information. Compact P3P policies may be sent along with a cookie request, for example, to notify the user-agent of the website’s policy with regard to cookie-based user identification and tracking. More details on P3P can be found in Cranor [26].

Since a website’s P3P document can be easily located and automatically parsed, many web browsers have integrated support for P3P. For example, Microsoft Internet Explorer 6 in combination with the PrivacyBird [13] plugin enabled users to explicitly set their privacy preferences and alter what types of information their browser reveals in response to a particular website’s privacy policy. For example, Privi-
cyBird may disable cookies for websites whose policies conflict with the user’s own privacy preferences.

2.4 The Limitations of P3P
While P3P offers a simple way for websites to specify their privacy policies in a manner that can be easily obtained and automatically parsed by user-agents, experience has shown that P3P is an imperfect solution. In our view, the most significant shortcomings in P3P are the following:

- **Enforcement**: P3P does not enforce the privacy policy that it describes – P3P is merely a format that facilitates automation. In some legal jurisdictions, a natural language privacy policy is legally binding. Websites may experience severe sanctions if they deviate from their policies as described. It’s unclear whether structured policy formats like P3P have the same legal implications as their equivalent natural language policies [28].

- **Few incentives for adoption**: It is well-known that P3P has not been widely adopted [27], in part because there have been few incentives driving sites to embrace the format. In the absence of privacy regulations that require websites to express their privacy policies in a structured format such as P3P, websites have little incentive to support P3P.

- **Expresses only privacy information**: Capturing information about an online service’s privacy policy is certainly important, but it describes only one dimension of the user’s interaction with an online service. Users may be interested in obtaining information related to a service’s terms of service or end-user license agreement, and P3P is not designed to describe this type of information. Further discussions of P3P’s limitations can be found in [20, 30].

It’s also possible that the slow acceptance and support for formats like P3P have historically been the result of insufficient interest in personal privacy protection from Internet users. However, in today’s complex web ecosystem with social networking sites storing an unprecedented amount of personal data, search engines aggregating personal information and retaining it indefinitely, and the increasing number of high profile data breaches [2, 16], privacy concerns and the protection of PII may be rising to the forefront of users’ minds.

2.5 Other Structured Policy Formats
Beyond P3P, other policy syntax models are being explored to suit various needs. In an effort to track, harmonize, and promote the various approaches is the W3C Policy Languages Interest Group [12] (PLING). Their goal is not to specify new policy formats, but to explore how existing ones can be more effectively applied.

The most extensible in active development that can be used to describe policies is the eXtensible Access Control Markup Language [10] (XACML), a standard managed by the Organization for the Advancement of Structured Information Standards [9] (OASIS). To some degree it could be perceived as being overly complex when compared to P3P, however XACML adds significant value as it is also includes defined syntax for use in managing access to specified resources.

3. SERVICE USAGE POLICY AUDITOR

Among the other formats being developed are those that strive to solve specific use cases. For example, there are two Kantara Initiative [8] Work Groups focused on specifying policy models: User-Managed Access [19] (UMA) and Information Sharing [7] (InfoSharing). Both are specifying formats for encoding privacy-respecting policies that can be used to support specific transactions. UMA is focused on delegating access to specified resources while InfoSharing is focused on supporting vendor-user data exchanges, though both efforts will depend on a ubiquitous policy-description format.

As UMA and InfoSharing are primarily interactive solutions designed to facilitate transactions, the DataPortability Project Policy Action Group [5] is specifying a format that can be used to express how a website intends to support the portability of personal data. As it is designed primarily to be informational, rather than transactional, it shares some of the same characteristics of P3P while being more focused on end-user comprehension than mechanical semantic identification.

2.6 A Case for Natural Language Policy Document Tracking

While prior proposals offer structured ways to convey policy information, unfortunately there exists no widely adopted structured data format for conveying information pertaining to an online service’s privacy policy, terms of service, or other important policies. Online services do, however, almost universally publish their various policies in human-readable, natural language text. Natural language policy documents are inherently difficult to automatically parse and semantically analyze by a computer. Further, even humans often struggle with the task of reading these complex documents [33]. Until a structured policy data format is deployed broadly, we propose that tools be developed to help users more effectively understand natural language policy documents. To this end, we next describe the design and implementation of a system that monitors natural language policy documents for changes and presents end-users relevant information about policies as they evolve.

Figure 1: The core components of the Policy Audit System

![Figure 1: The core components of the Policy Audit System](image)
change. In short, the system monitors changes in policy documents for a pre-defined set of online services and informs interested users of the changes.

Key design requirements of the system architecture are immediate applicability, extensibility, and privacy protection. While it would be easier to monitor usage policies that leverage a standardized approach (e.g., P3P), our design focused on providing immediate value by tracking currently available natural language policies. While this decision doesn’t preclude us from eventually tracking more semantically defined structured policies, it does enable us to provide immediate value.

**Policy Audit System components.** We separated the basic components in an effort to lay a strong foundation upon which more long-ranging functionality can eventually be built. Dividing the architecture along functional lines enables each component to be developed and grow independently. This also supports unforeseen innovation and expansion of the system’s utility. As illustrated in Figure 2, the system components are the following:

- **Policy Monitor:** The mechanism that actively monitors the known sites’ usage policies for changes.
- **Policy Library:** The repository storing active policies and their previous versions.
- **Policy Audit Clients:** Tools that provide end-users with information regarding the usage policies for the sites being visited.

### 3.1 Privacy-Preserving Design Considerations

An important consideration when designing the Policy Audit System was the need to take reasonable steps to protect the privacy of our users. This design consideration was driven by the anticipation that early adopters would be users more focused on privacy protection than the general user. In an effort to support these users, the overall architecture needed to be designed to balance the need for distributing accurate policy data, while also empowering users with options to minimize their personal privacy risk profile.

A common privacy concern to be addressed is that of user profiling, specifically without the knowledge of the user. This occurs when a server is able to relate a specific user to web pages they visit on one or more sites. The first step in constructing a profile for a user is to uniquely tag the user so they can be associated with the web pages they access. This is often done using identifying cookies or a static (or long-leased) IP address. Even when a user disables cookies and is sharing an IP address with others (in the case of a proxy), server access logs can contain enough information from common HTTP headers sent by the user’s browser to link, with reasonable accuracy, a specific user with their browsing history [3,11,23–25].

Users thinking about their privacy to this level are also expected to be concerned about a single site (i.e., the Policy Library) that could potentially log their queries for information regarding usage policies of the specific sites they visit. The potential threat is that a Policy Library may build a profile of a user’s personal web browsing habits by analyzing their queries, which is a severe violation of users’ expectations of privacy. We discuss our approach to mitigating this risk in the subsequent discussion of each component.

### 3.2 Policy Monitor (PM)

The Policy Monitor is a mechanism that tracks known policies published by specific websites. The Policy Monitor is built on the TOSBack service [18] developed by the Electronic Frontier Foundation (EFF) [6], modified to store copies of the tracked policies in the Policy Library.

Policies being tracked by the PM are loaded manually by a system manager who enters the URLs for the site usage policies and identifies the policy content within the published page. The PM then runs every 3 hours, retrieving each identified policy and comparing it to the most recent version in the database. If the policy has changed, a new snapshot of the current version is stored and the database updated accordingly. The results of the check are then output into the Policy Library, updating the appropriate files enabling Policy Clients to effectively access and display the information to end users.

The initial sites being tracked were determined by the Alexa top 50 most visited sites in the US [1] that publish policies accessible to the Monitor. We added to this initial list a number of policies for sites we felt to be of interest to our early adopters. Among the type of policies being tracked include end user license agreements, privacy policies, terms of service, terms of use, and others.

As there is no accepted standard for representing site usage policies in a common format (e.g., P3P), the policies being tracked by the PM are nothing more than published web pages. This means that the PM can effectively track any web page, comparing the differences over time, and the Policy Audit System relies on the manual identification of the policy type when it is entered into the PM to be tracked. While this does limit the ability for the PM to automatically detect and track unknown policies, it does provide the ability to track any published policy document.

### 3.3 Policy Library (PL)

The output of the PM is inserted into the Policy Library (PL), making the data available for public consumption. The PL stores the metadata about each tracked site and associated usage policies. The design of the PL is of a flat directory structure in an effort to empower services to mirror the data. The PL is the primary distinction between the Policy Audit System and the TOSBack service upon which the PM is based.

By exposing the tracked policy data via the PL, we are encouraging developers and other interested parties to leverage the information for their own purposes. The PL is a simple directory that is served via a web server consisting of:

- **Domains:** Services being tracked and associated metadata.
• **Snapshots:** Copies of each version of the tracked policies.

• **Change logs:** Easily consumable views of changes per run.

The PL is designed such that it can be accessible in two distinct modes: *interactive* or *download*. When being queried interactively, information regarding a single site usage policy, or all policies for a given site can be retrieved. Otherwise, the complete PL can be retrieved as a single archive file. For example, when querying the policies root, the result is a compressed archive of the entire directory as of the last detected change.

### 3.3.1 Mirroring the Policy Library

We also encourage replicating and re-publishing the PL on other servers (i.e., mirroring) in an effort to distribute the load, support unforeseen use cases, and to improve privacy protection. There are two classes of mirrors: certified (run by known organizations agreeing to manage them in a privacy-respecting manner), and anonymous (run by unknown organizations or users who are managing them on their own).

When considering the protection of a user’s privacy, it was important to enable the mirroring of the PL so that no single server was in a position to log all user access requests. If there was only one PL, it is possible that the server could log each request for policy data (potentially tagging each user with a reasonably reliable identifier). In such a centralized solution, the PL server could correlate the access logs to reconstruct a user’s surfing behavior relating to the tracked sites. By mirroring the PL, users can access any of the PL mirrors, minimizing the possibility that users’ surfing behavior can be reconstructed.

The “certified” mirrors take this one step further and publicly announce that they run their PL mirror in a manner that protects user privacy by:

- Hosting the PL mirror via a secure web service (i.e., using SSL/TLS).
- Maintaining a current secure certificate that can be publicly validated.
- Keeping minimal access logs, only as may be required for server maintenance.
- Frequently anonymizing, sanitizing, or deleting any access logs retained.

We are currently collaborating with the EFF and the Center for Democracy and Technology (CDT) [4] in the hosting of “certified” PL mirrors. Additional certified mirrors may be available in the future.

Further, users who are highly sensitive to privacy may opt to run their own PL mirror rather than access a version being hosted by a third party. In this case, they can maintain a local (or otherwise self-hosted) mirror by simply synchronizing their version with the master at periodic intervals. In this way, they are not exposing any access requests beyond simply downloading updated policy snapshots.

### 3.4 Policy Audit Client (PAC)

Once the tracked policy data is made available via the PL, a user interface is needed to effectively present the information regarding site usage policies to the users. We anticipate a number of PACs being developed to support various use cases. Rather than expecting users to visit a destination site that provides information about all policies being tracked, the PL enables PACs to be developed that “push” information about specific policies to interested users. Example “push” mechanisms supported by PACs may include RSS, ATOM, SMS, Twitter, e-mail, etc. so that end users can “subscribe” to changes in the usage policies for specific sites.

Another supported PAC use case would be for a site that leverages various Single Sign-On (SSO) mechanisms (e.g., OpenID, Facebook Connect, etc.). The site can develop their own PAC that queries the PL, displaying for their users links to the most current policies (or policy changes) related to their supported Identity Provider (IdP). Users may want to be aware that their choice of IdP comes with certain usage constraints, and the time for them to be notified of these constraints is when they are about to use their service.

### 3.4.1 Prototype Policy Audit Plugin

As a PAC proof-of-concept, we developed a Policy Audit Plugin (“Plugin”) for the Firefox web browser. The Plugin accesses the PL to alert the user when they visit a website that publishes a policy that the PM is tracking. The alert indicates whether or not the user has viewed the policy page(s) associated with the site. The user is able to easily view the policy page(s) from the alert icon displayed within the lower-right of their browser. If the policy page(s) changed since the last time they were viewed using the plugin, they are also presented with the ability to perform a “difference” comparison between the current version and the one they previously viewed.

The Plugin offers the following benefits:

- It alerts users in a consistent manner of the existence of tracked policies, rather than expecting the user to find where the specific site publishes their usage policies.
- It indicates the existence of tracked usage policies immediately upon visiting the site (when the policy often goes into effect), rather than at some possible later date when they may create an account on that site.
- It alerts users when tracked policies change, rather than relying on the user to remember to periodically check the policy manually.

Similar to the overall architecture of the Policy Audit System, the Plugin was designed to respect its users’ privacy. Included in the initial installation are the URLs for all of the “certified” PL Mirrors. In addition to ensuring a secure connection between the user’s browser and the PL Mirror to protect against inspection during transit, the Mirrors are accessed in random order to further obfuscate individual user access behaviors. Users who are highly sensitive to protecting their privacy can also enter the URL for their own locally hosted PL Mirror so that all queries performed by the Plugin go to a destination under their control. Concerned users who do not wish to host their own PL Mirror can use of an anonymizing proxy network such as Tor [29] in conjunction with the TorButton Firefox plugin to normalize HTTP headers [17] and conceal their identities from the PL Mirrors.
3.4.2 User Interface

The Plugin’s primary objective is to present the user with a simple, yet unobtrusive status interface to tell the user whether one or more policy documents have changed for the currently visited site. When the user visits a website with one of more tracked policies, the Plugin displays an indicator icon at the bottom of the browser window. As shown in Figure 3, the user may be presented with one of three different icons.

User visits site with no tracked policies. For websites with no tracked policies, the user is presented with a gray “X” indicating that the plugin has no information about any policy for the current site.

User visits site with a policy change. When the user visits a site with at least one tracked policy that the user has not reviewed or with policies that have changed since the user’s previous visit, the Plugin presents the user with a yellow exclamation mark. This indicates that the user’s attention is required to review a new or recently changed policy. Figure 3(a) shows an example policy change where the user is given an option of reviewing changes. If the user is interested in examining how the policy document has changed, they may view a side-by-side comparison with the textual differences highlighted, as shown in Figure 3(b).

User visits site with no change. When the user visits a site whose policies have all been read, and have not changed since the previous visit, the Plugin presents a green check mark to indicate that no action is required from the user.

4. FUTURE WORK

The initial goal of this project is to make site usage policies more visible to end users, especially when they change. This is, however, anticipated to be a foundational step upon which we can help build more effective methods of user engagement.

The end goal of this, and related work in which we are engaged, is to improve communication between service providers and their users by clarifying the policies governing their interactions. The Policy Audit System presented here is a step along this path as a method to more clearly present usage policies to users and track them over time. We anticipate, however, that while there will be an ongoing need for mechanisms such as this, the specific features it supports will continue to evolve.

4.1 Feature Support

Specific features we plan to support in subsequent versions of the Policy Audit System include:

- Policy Submission: A mechanism by which users and service providers can submit usage policies to be tracked.
- Policy Reviews: A mechanism that would allow external parties (authorized contributors or other users of the system) to contribute commentary to the PL regarding specific policy versions. These reviews would further explain, from various viewpoints, the pros and cons of the policy changes so users can make informed decisions.
- Agreement Tracking: Extending our system to track agreements similar to those being proposed by the User-Managed Access Work Group [19], DataPortability Project [5], ProjectVRM [15], and others.
- Feed and Push Notifications: While the directory structure of the PL makes it easy for programmers to distribute, mirror, and reuse the data, there are other potential methods for distributing the data we will endeavor to support as well. Among the methods being explored are feeds of recent changes in common subscription formats (e.g., RSS, Atom) as well as mechanisms to dynamically publish the information (e.g., PSHB, Twitter).

In addition to the features we foresee adding, we also intend to support third-parties that interact with the PM and PL. We anticipate that additional PACs will be built that access the PL, interacting with the policy data in various ways. We intend to continue supporting the Policy Audit System, reacting to collaborative developers, in a way to encourage adoption and open innovation.

4.2 Research Analytics

Beyond feature development, we intend to leverage the Policy Audit System as a functional illustration of the importance of service providers to effectively communicate their policies with their users. As such, we intend to use this as a tool to actively encourage the continued development and adoption of a formal, standardized process and syntax to encode and make available service usage policies.

In support of this effort, we understand the need for rigorous evaluation of the realistic needs of online services, their users, and the regulatory environments in which they operate. The Policy Audit Plugin PAC described above provides a compelling basis by which we can begin to gather significant data relating to user interaction with usage policies. Among the data points we would like to explore are:

- Normalized Policy Reading Patterns: Usage policies are currently published in a largely ad-hoc manner by service providers. While many provide policy links within the footer of a website page, that is not always
A user visits a site with a policy that changed since their previous visit.

When the user chooses to review changes, the newest version is displayed beside the previous version, with highlighted additions (in yellow) and deletions (in blue).

Figure 4: A user is notified about a policy change.

Correlation of Policy Reading with Events: As previous work has shown, users become more interested in site usage policies when they learn about an event that gives them reason to be concerned about how their personal data is being used and protected. Analyzing access logs for usage specific policies, and evaluating them for correlation with events (e.g., data breaches, media reports of identity theft, etc.) it may be possible to draw empirical evidence supporting direct as well as indirect increases of user interest in policies.

Evaluation of Structured Policy Formats: A problem faced by those interested in developing standardized policy formats is how they will be perceived once deployed. By leveraging an A/B testing paradigm, it will be possible to segment test groups within the Policy Audit System who are each provided with various policy formats. This side-by-side comparison between policy structure, presentation, and comparison of changes would work as a large-scale test bed for proposed standards.

These types of studies will be long-running and require the participation of multiple stakeholders. Research institutions, standards development groups, and advocacy organizations can use the Policy Audit System together with service providers and government regulators to find the most effective solutions. Key to the success of this model is participation of end users. Through the interaction with PACs such as the Policy Audit Plugin they should be engaged early in the research process to ensure truly realistic solutions emerge.

5. RELATED WORK

The Electronic Frontier Foundation’s TOSBack service monitors the policy documents for a well-defined set of websites [18]. TOSBack works by periodically fetching an online service’s policy document HTML source, extracting the relevant policy text using a specially crafted regular expression, and comparing the policy text to a previously recorded version of the policy text. If a change is detected, TOSBack presents a comparison of the current and previous versions of the text, highlighting the differences so the reader can easily identify precisely what changed. In doing so, the reader isn’t required to read the entire policy to recognize the additions or removals.

The system we present is based on TOSBack in its policy tracking functionality, but we introduce a few important extensions. Our system exposes a directory structure that allows arbitrary clients to easily fetch policy change data and present it. As an example client, we developed a Firefox plugin that enables the browser to automatically check the status of a website’s policy documents as the user visits the page. If a change has been detected, the user is presented with the policy changes (in the same manner as TOSBack) and can decide whether or not to continue to the website after studying the policy changes.

McDonald et al. study different privacy policy formats and their impact on users’ reading comprehension [33]. Relative to layered notices and P3P-summarized privacy policies represented in an expandable grid-like format [34] prepared by Privacy Finder, they find that natural language policies offer a lower rate of policy comprehension among a survey group. Unfortunately, given the limited deployment of P3P, it is unlikely that many website’s policy can be represented in the Privacy Finder’s easily-readable format. Other recent work has proposed a standardized privacy policy labeling scheme inspired by “nutritional labeling” that is now ubiq-
utious on food packaging [31]. As future work, we plan to explore the efficacy of this type of approach in comparison to others.

6. CONCLUSION

We present the system architecture of the Policy Audit System, a set of tools that automatically tracks changes in policy documents for a wide variety of online services. Since there is currently no standard and widely deployed machine-readable structured policy format, our system extracts the relevant text from human-readable natural language policy documents, detects salient changes in the policy text that are made over time, and presents users with a summary of the policy changes.

In an effort to increase the visibility of these policy changes to interested users, the notifications of policy changes are inserted directly into the user’s normal web browsing environment. This is achieved by alerting users of the changes via a browser plugin immediately when the user visits a website. Our system also empowers interested developers to leverage the collected policy information.

While this work is in the preliminary stages of development, we have ambitious plans to track a large number of policy documents across a wide variety of online services and help support the development of various applications that leverage the collected policy information.

7. REFERENCES


