Toward User Control over Information Access: A Sociotechnical Approach

Caleb Malchik
caleb.malchik@yale.edu
Yale University
New Haven, CT 06520, USA

Joan Feigenbaum
joan.feigenbaum@yale.edu
Yale University
New Haven, CT 06520, USA

ABSTRACT

We study the relationship between Web users and service providers, taking a sociotechnical approach and focusing particularly (but not exclusively) on privacy and security of personal data. Much conventional Web-security practice seeks to protect benevolent parties, both individuals and organizations, against purely malevolent adversaries in an effort to prevent catastrophic events such as data breaches, ransomware attacks, and denial of service. By contrast, we highlight the dynamics *among* the parties that much conventional security technology seeks to protect. We regard most interactions between users and providers as implicit negotiations that, like the interactions between buyers and sellers in a marketplace, have both adversarial and cooperative aspects. Our goal is to *rebalance* these negotiations in order to give more power to users; toward that end we advocate the adoption of two techniques, one technical and one organizational. Technically, we introduce the Platform for Untrusted Resource Evaluation (PURE), a content-labeling framework that empowers users to make informed decisions about service providers, reduces the ability of providers to induce behaviors that benefit them more than users, and requires minimal time and effort to use. On the organizational side, we concur with Gordon-Tapiero *et al.* [19] that a collective approach is necessary to rebalance the power dynamics between users and providers; in particular, we suggest that the *data co-op*, an organizational form suggested by Ligett and Nissim [25] and Pentland and Hardjono [28], is a natural setting in which to deploy PURE and similar tools.

CCS CONCEPTS

• **Information systems** → **Information retrieval**; *World Wide Web*; *Social tagging systems*; Open source software; • **Security and privacy** → *Human and societal aspects of security and privacy*.

KEYWORDS

user control, data co-ops, content filtering

ACM Reference Format:
Caleb Malchik and Joan Feigenbaum. 2022. Toward User Control over Information Access: A Sociotechnical Approach. In *New Security Paradigms Workshop (NSPW '22), October 24–27, 2022, North Conway, NH, USA.* ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3584318.3584327

NSPW '22, October 24–27, 2022, North Conway, NH, USA

ACM ISBN 978-1-4503-9866-4/22/10.
https://doi.org/10.1145/3584318.3584327

1 INTRODUCTION

Social networks, online-commerce platforms, and other Internet-based companies derive tremendous value from observing and influencing users and from collecting and analyzing users' behavioral data. Recent years have seen a groundswell of public objection to the lack of accountability on the part of large, Internet-based service providers. In this paper, we propose a two-pronged, sociotechnical approach to increasing accountability of these companies.

By "a sociotechnical approach," we mean an approach to the design and implementation of systems affecting people's everyday lives that can only be understood and improved in a multidisciplinary fashion: Social and technical aspects must be brought together and treated as interdependent parts of a complex system. The sociotechnical approach is a natural one to take when trying to increase users' control over and service providers' accountability for user's online experience, because social mechanisms intended to increase users' control will need technological instantiation, and technical mechanisms are likely to be adopted only if they are enforced legally, incentivized financially, or supported by social trends or norms.

1.1 The PURE Label Framework

Our technical contribution is PURE—the Platform for Untrusted Resource Evaluation—a content-labeling framework that empowers users to make informed decisions about service providers, reduces the ability of providers to induce behaviors that benefit them more than users, and requires minimal time and effort to use. In this context, an "untrusted" resource is one that is offered by a provider that might mislead users about its properties, typically because of financial incentives but often for political, social, psychological, or other reasons. In order to inform a user about the properties of online resources and to guide him towards resources that suit his preferences, PURE aggregates and processes *labels* provided by a variety of sources. Client-side content-discovery tools then present online resources in a way that favors this user's priorities and minimizes the harms that he deems most important.

PURE does *not* expect every individual user to formalize his values, both positive and negative, regarding online resources in the logic and notation of a particular labeling system. Rather, PURE is based on the assumption that sources who agree upon a specific property of one resource are likely to agree in the case of another property and resource. Users can decide which sources to subscribe to and sort them in order of trustworthiness or, crucially, can delegate this task to data co-ops that embrace their values. They can also supply feedback to co-ops about whether online resources actually have the properties that they claim to have, thus enabling the

co-ops to improve their label-processing systems and give better guidance to their members about which online resources to use.

Properties of PURE that we believe are key to user empowerment include:

- It is a *grassroots* solution: It enables users to act with like-minded peers to avoid and resist online harms without depending on large, powerful organizations with whom they may have conflicting interests.
- It is a *lightweight* solution: It relies on client-side tools that are easy to use and that cause minimal disruption to typical Web users' experience.
- *Flexibility*: The client-side tools that we present do not require users to abandon popular online services to which they have become accustomed. Our tools can be used in conjunction with existing services; indeed, they allow people to use those services in a more deliberate and informed manner and to become aware of more privacy-respecting and generally better alternatives.
- *Extensibility*: Our solution establishes a basis for more radical changes to Internet protocols. It is well suited to alternative transport protocols and document formats designed with privacy and user control in mind.

1.2 The Data Co-op Organizational Framework

The goal of *data co-ops*, as conceived by Ligett and Nissim [25] and Pentland and Hardjono [28], is to empower users in their online interactions with providers by facilitating collective action. During the past decade, a number of related, smaller-scale initiatives have focused on issues like collecting higher-quality data, (re-)gaining individual control over personal data, generating rich data for research purposes, *etc.* Each of them addressed one or more specific shortcomings of the status quo, *e.g.*, privacy risks, unrealized potential of data, individuals' receiving limited value in exchange for their behavioral data, and lack of opportunities for public governance of digital creation. By contrast, data co-op research seeks more broadly to address, in a sociotechnologically comprehensive fashion, users' disadvantages in the current environment.

In our formulation, a data co-op is a membership organization that provides client software; a means of resource discovery; and technical, legal, and social support. Co-ops could also provide means for users to communicate or publish content, such as digital identities and hosting resources. Thus, in the short run, services and client software provided by data co-ops as we conceive them would advance the goals of increased user privacy and more widely distributed value creation that were put forth in [25, 28]. Longer-term goals are discussed in Sec. 4.

There are many possible revenue models for such an organization, the simplest of which is to have the members pay dues. We expect co-ops to exhibit great diversity in the online harms they focus on mitigating, the number of members they have, and the range and complexity of member services that they provide. Some will be able to support their activities with very modest dues and/or to allow members to substitute service to the organization (*e.g.*, regular labeling of content and websites) for all or part of their dues payment.

1.3 Use Cases

Anything that can be labeled and verified by multiple sources may be targeted by data co-ops with PURE. In this way, co-ops can encourage or discourage certain practices or empower users with diverse needs and desires to improve their experience.

We suggest three categories of real-world intervention that could be pursued using our sociotechnical approach.

1.3.1 Information Quality and Information Risks.

Labels based on the information content of a resource could be used to encourage balanced reporting of news, to discourage misleading or false information, and to identify content that may be particularly troubling to certain people.

Dominant platforms already engage in content moderation with the goal of reducing misinformation and improving information quality, but the extent and specific targets of these efforts are not made public, which ironically reduces public trust in these efforts. PURE labels identifying the specific issues with a piece of content could accomplish much of the same moderation with greater transparency and accountability. Data co-ops could coordinate a comprehensive effort to make existing information-quality efforts by platforms transparent in a way that is compatible with PURE.

Social-media users often provide content warnings for posts that may trigger acute anxiety in people who have experienced trauma. PURE labels for these warnings would enable people who would normally avoid reading content with certain warnings to filter it out completely.

1.3.2 Client-Software Diversity and Accessibility.

PURE and data co-ops could be used to facilitate the use of alternative client software, including browsers with less market share as well as privacy-oriented browser extensions and screen readers for the blind.

With the market dominance of Google Chrome and Chrome-based browsers [1], there is concern over web-browser monoculture, with some citing it as a security concern [21, 26] and others lamenting the supplanting of "consensus and cooperation" in favor of "corporate rule" [32]. Browser monoculture is exacerbated by the possibility that certain websites fail to work in all browsers or even all major browsers. This encourages people to use the most dominant browser because it is most likely to work for any given web page. The same dynamic applies to browser extensions that take anti-tracking measures such as NoScript [3] and Privacy Badger [4].

Blind users are often faced with imperfect support for screen readers on web pages. Even pages that work with screen readers may provide a user experience of reduced usefulness as compared with that of sighted users [15], motivating more radical approaches such as the "command line editor browser" edbrowse [2], which may provide an enhanced user experience at the cost of reduced compatibility.

PURE attributes indicating whether a page is compatible with certain client software (such as *noscriptcompat*, referenced in Sec. 2.2) could filter or demote pages that are not compatible with a user's chosen software, eliminating constant nudging to use software that is better supported and pressuring publishers to avoid practices that create a dependence on any particular client software.

The goal of pressuring publishers to support various client software can be furthered by a data co-op. Knowing that others in one's data co-op are using the same client software and PURE labels to filter incompatible pages could make it easier to bear the unseen cost of missing out on certain content. A data co-op could also take a more active role, contacting strategically chosen publishers to request a change in page templates with the promise of increased traffic from co-op members.

1.3.3 *User-hostile Design Patterns.*

Labels could be employed to identify content that employs "dark patterns." This term was coined by Harry Brignull [12] to describe deceptive user-interface elements designed to trick the user into actions that benefit the provider of the interface more than the user, wherein "user value is supplanted in favor of shareholder value" [20]. PURE and data co-ops could discourage dark patterns and other annoying or user-hostile design elements. We discuss this use case in depth in Sec. 3.1.

1.4 Context and Motivation

1.4.1 *Implicit User-Provider Negotiations.*

When a user interacts with an Internet service under typical market conditions, the user and the provider both want to maximize the value they derive from the interaction while minimizing costs. The interests of the two parties are aligned in some aspects, such as the desire to meet the needs of the user so that she continues to use the service. In other aspects, their interests conflict; for example, in an ad-supported service, the provider wants to maximize the data collected about the user, the time and attention spent on ads, and the capacity to influence user behavior, but the typical user wants to minimize these things.

The adversarial component of this relationship leads to an implicit negotiation: Users want to fulfill their needs while offering the provider only enough value to sustain the service. The provider wants to maximize value extraction, while offering users only enough value to motivate them to continue to use the service. The strong market positions of many of today's providers, together with the relative indifference and atomization of users, causes these interactions to tend toward the latter extreme. For example, the Platform for Privacy Preferences (P3P) [14] sought to make a class of these negotiations explicit and to empower users to delegate relevant decisions to their browser software, but it failed to gain widespread adoption because browser implementers did not support it adequately. Our position is that the type of *collective action* that is endorsed in [19, 25, 28] can be more effective than individual, purely technological action in strengthening the position of users.

Negotiations between users and service providers resemble those between buyers and sellers and between employers and employees (buyers and sellers of labor). In the same way that consumer cooperatives and labor unions can shift negotiations in favor of consumers and employees, data co-ops could shift these implicit negotiations over data, attention, and services in favor of users, leading to services that are less invasive, manipulative, and addictive, as well as more useful and reliable.

1.4.2 *Types of Intervention and Prospects for User Control.*

Efforts to address the harms of mass data collection have come in many forms: technical interventions such as Tor [16] and Pretty Good Privacy [18], legal interventions such as the European Union's General Data Protection Regulation, and interventions by advocacy organizations such as the Electronic Frontier Foundation's "report cards" and the Free Software Foundation's promotion of free software. These categories are not strictly disjoint: Any non-technical intervention requires technical efforts to implement, and any technical intervention requires non-technical efforts to support its use.

It is helpful to examine the free-software norm in view of the implicit-negotiation dynamic described above. Free software (*i.e.* software that allows unrestricted modification and redistribution of its source code) offers a protective mechanism against anything that the user would not want a program to do, because users can modify the program or pay someone to modify it for them. However, this requires nearly boundless resources (while users do not even have time to read the terms of use for most services), and modifying client software can introduce incompatibilities with the server.

The shortcomings of the free-software norm are illustrated by the evolution of the Web. As the Web grew more complex and more dominated by huge companies, the complexity of browsers reached a point at which even a free browser like Firefox is beyond the control of users, because a large team of developers is required to make changes without breaking things. Even if this barrier were overcome, many websites require JavaScript, which is often obfuscated. Even if a user examines the JavaScript code on every website that she visits and modifies it to prevent needless data collection or manipulative user-interface elements, these changes could break the website's functionality.

The evolution of the Web also demonstrates that effective control over what data are sent to the server requires regulation of the entire client program, or of the data it receives, and not just the network protocol. HTTP was designed for simple hypertext, which lacks many of the data-collection capabilities that modern websites employ, but HTTP can still transport tracking scripts that are run by a browser engine and report behavioral data back to the server (also via HTTP). Designing the protocol for simple hypertext did not prevent these innovations.

For a data co-op or coalition of co-ops to maintain control over the client program, the client code that is specific to a service provider should be minimized. Users are likely to want to continue using established sites for search, online commerce, and other essential services and thus unable in the short run to escape behavioral-data collection and other harms altogether, but labeling can inform users, raise their awareness of harms, and make it easy for them to preferentially use less harmful sites whenever possible. This approach also provides a path to gaining control over the client software by filtering out resources that require software developed by unaccountable third parties, as discussed in Sec. 1.3.2.

1.4.3 *Related Work on Grassroots Action by Users.*

The potential effectiveness of data co-ops and our particular approach to them can be understood and evaluated using the data-leverage framework of Vincent *et al.* [34], the goal of which is to

"highlight new opportunities to change technology company behavior related to privacy, economic inequality, content moderation, and other areas of social concern." Three levers that are available to users are identified in [34]: *data strikes*, in which users withhold or delete data to reduce the efficacy of an organization's data-dependent technologies; *data poisoning*, in which users insert inaccurate or harmful data into an organization's data-dependent technology; and *conscious data contribution* (CDC), in which users give their data to organizations they support.

Data co-ops can provide their members with data leverage. In particular, they can manage labels associated with websites, thus effectively subjecting certain sites to boycott; in this way, co-ops can coordinate data strikes, data poisoning, and CDC by their members. Labels also allow this point of leverage to be used more fluidly and subtly by steering users away from misbehaving sites and towards more favored ones.

Many works, notably those of Posner and Weyl [29] and Arrieta-Ibarra *et al.* [9], promote the concept of "data as labor." Currently, corporations that monetize users' data view those data as their property that *they create* by providing services to people who use them willingly. Adherents to the data-as-labor school of thought view data as valuable products that *users create* and that corporations profit from. Their view leads naturally to the goal expressed here and in [34]: Rebalance the relationship between the users who create data and the corporations that profit from them so that users have more knowledge about which data are collected, more control over how data are monetized, and more ability to reap the rewards of data-dependent commerce. Data co-ops may thus be viewed as analogous to labor unions, which serve precisely this function in negotiations between workers and their employers. This analogy was drawn explicitly by Pentland and Hardjono [28]. Similarly, Posner and Weyl suggest the formation of "data unions," with a focus on monetary compensation for data.

Data co-ops involve both peer evaluation of online resources and collective action by users. The use of peer evaluation in the absence of a formal collective has also been explored. For example, Jahanbakhsh *et al.* [22] demonstrated the utility of individual users' providing accuracy assessments of social-media posts before sharing them. Their tools locate intelligence and processing on machines separate from the providers' servers, and they emphasize *lightweight* intervention and ease of use. PURE also exemplifies these principles.

1.5 Paper Outline

Sec. 2 contains the design, implementation, and performance analysis of the PURE framework. In Sec. 3, we explain how data co-ops can use PURE and why we believe they offer a realistic path to user empowerment and provider accountability. Finally, in Sec. 4, we present future directions, some concrete and some long-term and highly speculative.

2 PLATFORM FOR UNTRUSTED RESOURCE EVALUATION

PURE is a framework for determining whether or not certain attributes apply to certain information resources based on client-side processing of labels from various sources. A PURE label is defined by:

$$\langle \textit{source}, \textit{attribute}, \textit{value}, \textit{name}, \textit{type} \rangle$$

where *value* can be 0 or 1 and *type* can be "specific" or "generic."

If *value* is 1 and *type* is "specific," this label expresses that *source* claims *attribute* applies to the resource identified by *name*. If *value* is 0, it expresses the negation. If *type* is "generic," the statement also applies to any resource for which *name* is a prefix.

PURE is based on the assumption that sources who agree on whether an attribute applies to a resource are more likely to agree in the case of another attribute and resource. Users define which sources to subscribe to and sort them into *tiers* according to their trustworthiness (or may delegate this task to an institution such as a data co-op). Users may also produce their own labels, and would typically be the only source occupying the highest tier for their configuration. Trust tiers provide automatic accountability for sources in lower tiers, enabling the use of relatively unfamiliar or unvetted label sources, including *e.g.* labels from publishers about their own publications.

We do not define a format or transport protocol for PURE labels; PICS [30] and the deprecated W3C DSig recommendation [13] offer two possible instantiations. PURE is distinguished by restricting the statements that can be made about a resource to attributes applying or not, in order to enable simple and efficient processing of conflicts between labels.

2.1 Label Processing

PURE labels are processed on the client side to maximize the autonomy of users and to reduce the transmission of user data. The processing computes two quantities: the reputation of a given source and the expectation of the value for a given attribute and name.

Reputation is a measure of the degree of agreement between a source and other sources in higher tiers, from 0 to 1. Let $\mathcal{L}_j$ be the set of tuples (i,k) such that source j has given a label for name k and attribute i and let v_{jik} denote the value for that label. Let $x = \mathrm{Exp}(i,k,t-1)$, to be defined further below. Then the reputation of a source j with tier t can be defined roughly as:

$$\mathrm{Rep}(j) = \frac{\sum_{(i,k)\in\mathcal{L}_j} |v_{jik} - \lfloor x + 0.5 \rfloor| \cdot |x - 0.5|}{\sum_{(i,k)\in\mathcal{L}_j} |x - 0.5|}$$

If the denominator is zero, meaning higher tiers are agnostic for the attributes and names labeled by j, then the reputation is 1. Note that tiers are identified by numbers, with lower numbers denoting tiers of higher priority ("higher tiers").

Expectation is a measure of the likelihood that an attribute applies to a name, from 0 to 1. It is based only on labels from sources in the highest tier that is not agnostic for the given attribute and name (a tier may contain relevant labels but still be agnostic if the labels disagree evenly with each other). For expository purposes, we have defined the function with a tier parameter t indicating the lowest tier to use. Let $\mathcal{S}_{tik}$ be the set of sources in tier t with labels for attribute i and name k. Provided that there is no higher tier $t' < t$ such that $\mathrm{Exp}(i,k,t') \neq 0.5$, the expectation for attribute i, name k, and tier t is:

$$\mathrm{Exp}(i,k,t) = \frac{\sum_{j\in\mathcal{S}_{tik}} v_{jik}\mathrm{Rep}(j)}{\sum_{j\in\mathcal{S}_{tik}} \mathrm{Rep}(j)}$$

```
foreach (src, attr, val, prefix) for which the label ⟨src, attr, val, prefix, generic⟩ exists do
    foreach name for which some label ⟨_, _, _, name, _⟩ exists and prefix is a prefix of name do
        create the label ⟨src, attr, val, name, virtual⟩
foreach tier t from 0 to t_max do
    foreach source src with tier_of(src) = t do
        if t = 0 then
            rep[src] := 1
        else
            n := d := 0
            foreach (attr, val, name, type) for which the label ⟨src, attr, val, name, type⟩ exists do
                if type = virtual then
                    truth := exp_nonvirt[name][attr]
                else
                    truth := exp[name][attr]
                if truth is unset then
                    continue
                certainty := 2 * |truth − 0.5|
                n += certainty * |⌊truth + 0.5⌋ − val|
                d += certainty
            rep[src] := max(1 − 2n/d)
    foreach (attr, name) for which some label ⟨_, attr, _, name, _⟩ exists do
        n := d := n_nonvirt := d_nonvirt := 0
        foreach (src, val, type) for which the label ⟨src, attr, val, name, type⟩ exists and tier_of(src) = t do
            n += rep[src] * val
            d += rep[src]
            if type ≠ virtual then
                n_nonvirt += rep[src] * val
                d_nonvirt += rep[src]
            if type ≠ specific then
                n_nonspec += rep[src] * val
                d_nonspec += rep[src]
        if d ≠ 0 and exp[name][attr] is unset then
            exp[name][attr] := n/d
        if d_nonvirt ≠ 0 and exp_nonvirt[name][attr] is unset then
            exp_nonvirt[name][attr] := n_nonvirt/d_nonvirt
        if d_nonspec ≠ 0 and exp_nonspec[name][attr] is unset then
            exp_nonspec[name][attr] := n_nonspec/d_nonspec
```

Algorithm 1: Initial processing to compute reputations for all sources (in the map *rep*) and some expectations (in the map *exp*).

If the denominator is zero, meaning no reputable source in tier t (or any higher tier) has given a label for i and k, then the expectation is 0.5.

We have given simplified definitions for reputation and expectation; a more precise illustration of how these values are computed, including the treatment of generic labels, is given in Algorithms 1 and 2.

In Algorithm 1, the mutual recursion between the functions above is handled by setting the reputation for the highest tier first (it is always 1), then computing the expectation for names and attributes that have been labeled by sources in that tier. Knowing these values allows the reputations and expectations from the next tier down to be computed, and so on.

Generic labels present a challenge because they raise the question of how agreements and conflicts should be handled when computing reputations: If a tier 1 source makes a generic assertion that overlaps with one from tier 0, and has the same attribute but disagrees on the value, how much should that diminish the reputation of the tier 1 source? Because generic labels refer to an unbounded range of names, one could argue that they should be given infinite

```
function expectation(name, attr) is
    if some label ⟨_, _, _, name, _⟩ exists then
        if exp[name][attr] is set then
            return exp[name][attr]
    else
        let prefix be the longest prefix of name for which
          some label ⟨_, _, _, prefix, _⟩ exists
        if exp_nonspec[prefix][attr] is set then
            return exp_nonspec[prefix][attr]
    return 0.5
```

Algorithm 2: Deriving expectation for any name and attribute from values pre-computed by Algorithm 1.

```
function adjustment(name) is
    r := 1
    foreach attribute attr in user policy do
        if attr is favored then
            q := expectation(name, attr)
        else
            q := 1 − expectation(name, attr)
        if q ≥ 0.5 then
            r *= 2q
        else
            r *= 0.5 / (1−q)
    return r
```

Algorithm 3: Computing the factor by which to adjust the relevance score of a search result in PURESearch.

weight, but this would discard any information from specific labels. Another approach is to use names incidentally present in other labels to determine the weight of an agreement or disagreement: if we have collected many names that are covered by a pair of generic labels, this suggests that there are many real resources affected by the generic assertions, so they should be given more weight. But this makes the reputation of sources dependent on labels from lower tiers, which may not be desirable, and at best creates ambiguity in the semantics of tiers. Thus, we give a comparison between two generic labels the same weight as that between two specific labels, or between one specific and one generic label.

To accomplish this, we create a "virtual" label for each name we have collected that is covered by a given generic label. These virtual labels are treated like any other label, except that we compute a non-virtual expectation value alongside the standard expectation, based only on specific or generic labels whose name matches the name under evaluation (not just a prefix). Then in determining reputations, we only measure a virtual label against the non-virtual expectation from higher tiers. This prevents high tier virtual labels which are created due to lower tier labels from affecting the reputations of sources in intermediate tiers.

At the end of Algorithm 1, the map *exp* contains expectation values for known names, for attributes that have been labeled for that name (or for a prefix of that name by a generic label). Generic labels make it possible to derive expectation values for an unlimited range of names; to accomplish this we compute a non-specific expectation value including information from generic and virtual labels. This is precisely the information that applies to any names which share a prefix with a given known name, so we use it in Algorithm 2 to derive expectation values for names we haven't seen.

We have written a program called `pure rep` which computes reputations according to Algorithm 1 and prints them out, in order to study the properties of the most complex and resource intensive component of PURE. The implementation reflects the goals of resilience and minimizing the necessary trust in software providers: It is written in approximately 300 lines of C and designed to be distributed under a free license as source code which is re-compiled when the configuration is changed. The simplicity and distribution method allow a small group of users to continue to maintain and use the system even if the original distributor becomes unavailable or is otherwise compromised. These properties also maximize the trust that users can have in the software by minimizing the cost of auditing or changing the code.

One must also trust the language implementation used to compile or run a program; the choice of C ensures that users have the option of many different compilers, including some that are especially simple and relatively easy to audit [6, 10, 17]. (We do not suggest that a typical user would audit the compiler for their client software, but pushing these ideas as far as possible maximizes the trust in the software, which is especially important for a core algorithm that enforces expressed ideas about trust.)

The `pure rep` program can be relatively simple because PURE makes minimal assumptions about what can be a label source and which attributes can be used. This also makes PURE flexible and able to support diverse usage patterns, as discussed in Sec. 2.3.

Performance measurements for `pure rep` are presented in Sec. 2.5.

2.2 Re-ranking Search Results with PURE

The promise of PURE to empower users in the domain of information access lies in the ability to label resources with attributes that users care about and to direct usage towards certain resources and away from others. To illustrate this, we have implemented a proof-of-concept interface called PURESearch, which re-ranks search results according to PURE expectation values and user preferences.

PURESearch uses a simple instantiation of PURE labels, with label sources corresponding to URLs which are queried for label records. Label records are returned as a text file, each line consisting of three tab-separated fields: a string attribute, a boolean value, and a URL naming the resource that the label is about.

User preferences are expressed as a list of favored attributes to be promoted in search results and a list of disfavored attributes to be demoted (the current prototype does not support generic labels).

When the user makes a PURESearch query, results are fetched from an instance of Searx [7], including relevance scores that determine the default ordering of results. These relevance scores are

multiplied by some adjustment value, computed according to Algorithm 3, and the results are reordered by the new adjusted scores (see Figure 1).

In Algorithm 3, q represents the *favorability* of the named resource with respect to attribute $attr$, so that a low expectation of a favored label yields the same effect as a high expectation of a disfavored label with the same level of uncertainty. The running product is multiplied by $2q$ for a favorable assessment and $\frac{0.5}{1-q}$ for an unfavorable assessment, so that an unfavorable assessment cancels out a favorable assessment with the same magnitude. This algorithm is designed for simplicity and generality; more complex algorithms may provide greater utility for specific use cases.

When the user clicks on a result, she may open a sidebar to inspect the labels for that result and enter her own (see Figure 2). This makes it possible for the user to correct inaccurate or inadequate labels if a result seems out of place or should have been filtered out.

PURESearch demonstrates the "lightweight" nature of the PURE approach: users may follow the same usage patterns to which they are accustomed, and may engage with PURE labels at their option but without feeling pressure to pay them any attention. The largest change experienced by users is the move to new content-discovery interfaces, which can be made familiar and functionally equivalent to popular interfaces supplied by dominant information services.

The source code for `purerep` as well as a proof-of-concept implementation and self-guided demonstration of PURESearch can be found at `https://cs.yale.edu/homes/cmalchik/pure/`.

2.3 Sources, Attributes, and Names: Flexibility and Considerations

PURE presents a simple and flexible interface on which to build applications such as PURESearch: sources, attributes, and names can be almost anything, and the effectiveness of PURE depends on these elements having certain properties.

The core assumption of PURE is that agreement between sources for a particular name and attribute implies these sources are more likely to agree in the case of a different name and attribute (and conversely for disagreement). When certain sources are more trustworthy as a base assumption (*e.g.* the user trusts himself or his data co-op), this makes it possible to draw inferences about the trustworthiness of other sources.

Sources in PURESearch are represented by URLs, an approach which relies on the assumption that the content at a source URL is controlled by a single conceptual authority. The binding between a source and its labels is verified by fetching labels from the source URL. An alternative approach would be to represent sources as public keys, distributed via some infrastructure, with labels produced by that source being signed by the corresponding private key. This would decouple the source of a label from its method of distribution, possibly making labels easier to distribute and thus more available to users.

Sources should be added to a PURESearch instance with the understanding that the source can affect the ranking of pages that appear in upstream results and that are not evaluated by sources in higher tiers. A malicious source may label an unpopular site (which is unlikely to have been evaluated by another source) with various inaccurate negative labels in order to demote it in results and make it even less likely to be seen. The malicious source may also inflate its reputation relative to other sources in its tier by copying labels from a higher-tier source. The best defense against these attacks is to only add sources that the user trusts to some degree, such as friends or organizations that have something to lose if such an attack is detected.

Resource names in PURESearch are also represented by URLs. This relies on the ubiquitous but shaky assumption that certain URLs refer to particular resources that won't change over time. In reality, the page pointed to by a URL may disappear or change over time, so that labels created at different times may refer to different resources. This could be remedied imperfectly by including a timestamp in each name, perhaps limited to the year and month, so that labels for pages made in the same month are assumed to refer to the same page. A more durable long-term solution would be to use some infrastructure that binds a name to a hash of the data, such as Named Data Networking (NDN) [35].

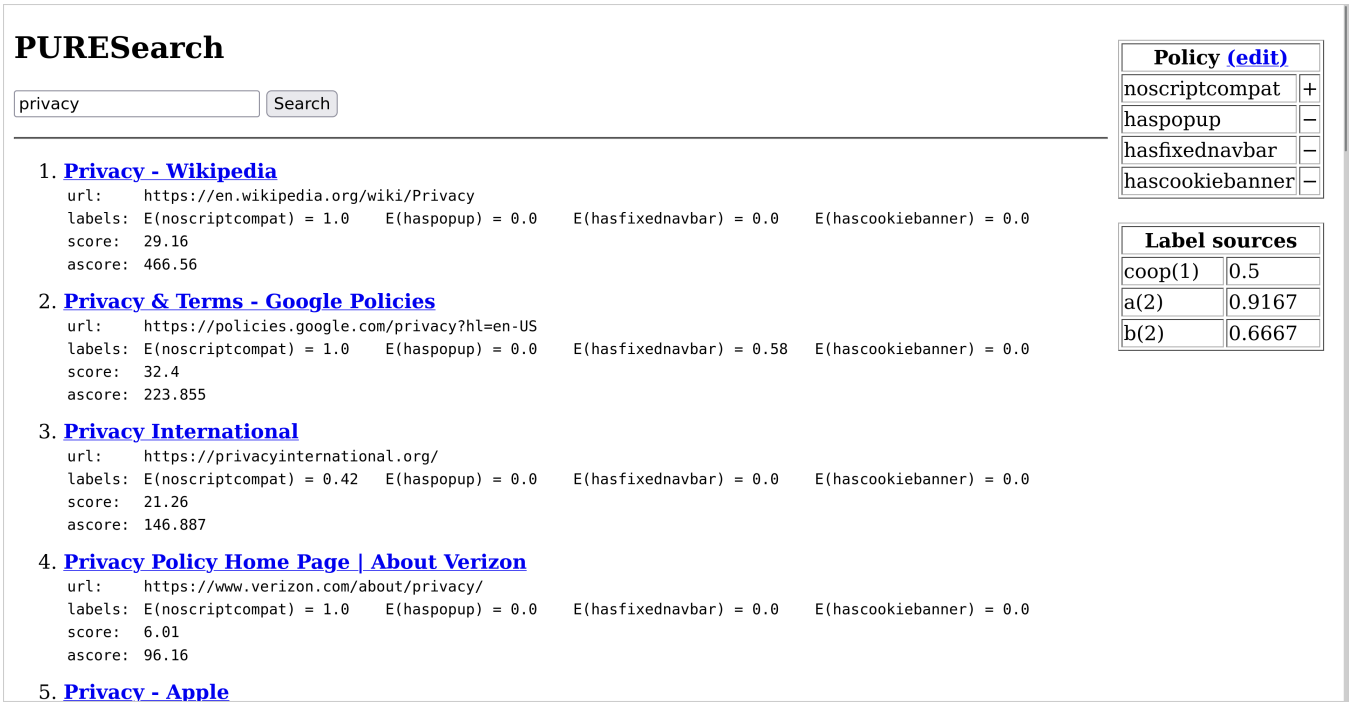
Because reputations in PURE refer to sources without regard to the attribute being named, sources and attributes should be defined such that being trustworthy for certain specific attributes plausibly means that a source is trustworthy in general. Attributes that best meet this requirement refer to properties that can be objectively verified, so that a label that the user disagrees with indicates dishonesty or incompetence on the part of the source, rather than a simple difference of opinion. Of course, objectivity is a flexible concept and attributes which leave some room for debate may be important to users. To prevent disagreement on subjective attributes from affecting the reputation of a source, labels may be pre-processed to rename sources in labels that refer to such attributes.

The example attributes we use to illustrate PURESearch (pictured in Figures 1 and 2) are unstructured strings. If PURE is to gain wide adoption, it will be important for different label producers to agree on the names of attributes. It may be helpful to impose some structure on attribute names, to avoid labeling slightly different properties with the same attribute and to make it easy for label producers to determine which attribute name to use for a given property.

Standards and conventions for sources, attributes, and resource names should be determined carefully and with input from various stakeholders, to prevent ad hoc and mutually conflicting schemas from fragmenting the PURE ecosystem.

2.4 Relation to Recommender Systems

It is important to note that, although PURE draws on the extensive bodies of work on systems that enable users to rate, rank, or review online content and services, it is a different type of system from these earlier ones. Recommender systems [23] are widely deployed in social media and entertainment platforms to predict which content will engage the user most effectively. At first glance, PURE might appear to be just an unusual type of recommender system that maximizes the user's control over what is "recommended." However, typical recommendation systems employ either or both of two broad approaches: content-based filtering and collaborative filtering. Although both are similar in some ways to the PURE approach, neither applies in a conventional way.



PURESearch

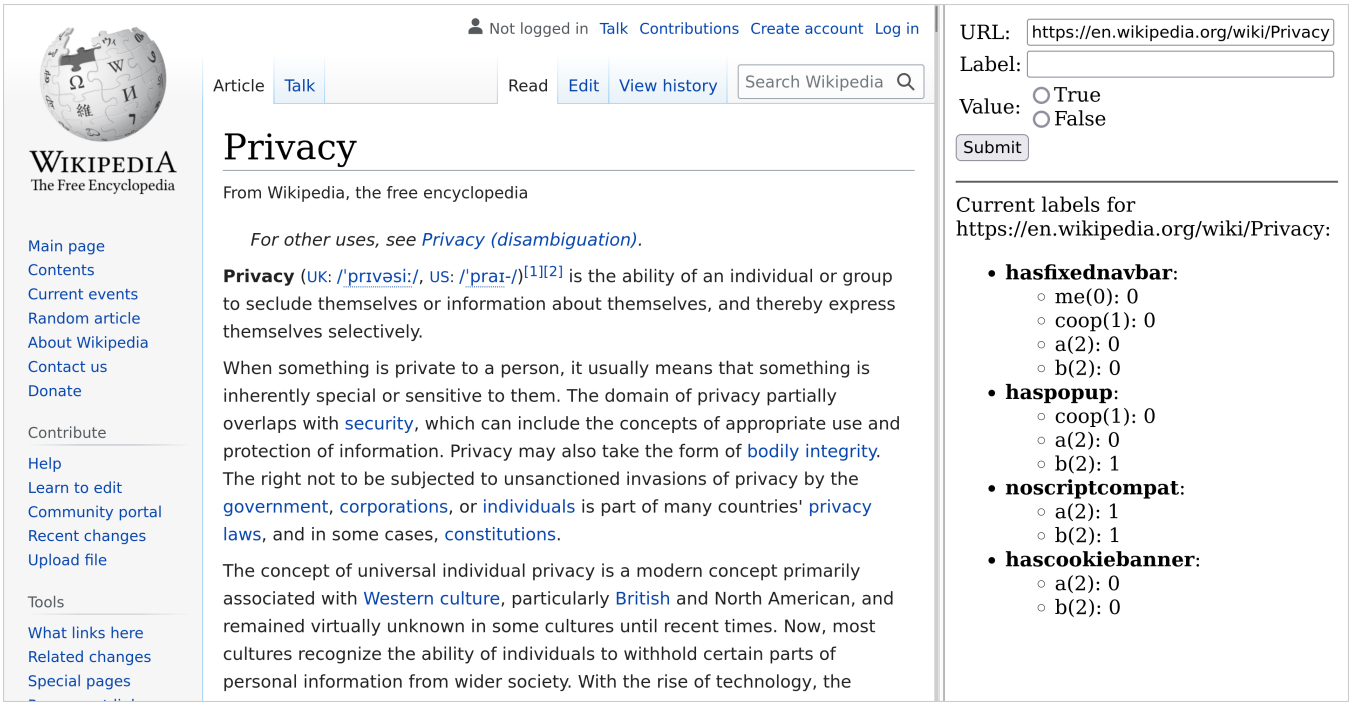
privacy Search

1. **Privacy - Wikipedia**
   url: https://en.wikipedia.org/wiki/Privacy
   labels: E(noscriptcompat) = 1.0 E(haspopup) = 0.0 E(hasfixednavbar) = 0.0 E(hascookiebanner) = 0.0
   score: 29.16
   ascore: 466.56
2. **Privacy & Terms - Google Policies**
   url: https://policies.google.com/privacy?hl=en-US
   labels: E(noscriptcompat) = 1.0 E(haspopup) = 0.0 E(hasfixednavbar) = 0.58 E(hascookiebanner) = 0.0
   score: 32.4
   ascore: 223.855
3. **Privacy International**
   url: https://privacyinternational.org/
   labels: E(noscriptcompat) = 0.42 E(haspopup) = 0.0 E(hasfixednavbar) = 0.0 E(hascookiebanner) = 0.0
   score: 21.26
   ascore: 146.887
4. **Privacy Policy Home Page | About Verizon**
   url: https://www.verizon.com/about/privacy/
   labels: E(noscriptcompat) = 1.0 E(haspopup) = 0.0 E(hasfixednavbar) = 0.0 E(hascookiebanner) = 0.0
   score: 6.01
   ascore: 96.16
5. **Privacy - Apple**

Policy (edit)	
noscriptcompat	+
haspopup	–
hasfixednavbar	–
hascookiebanner	–

Label sources	
coop(1)	0.5
a(2)	0.9167
b(2)	0.6667

Figure 1: PURESearch results for "privacy," including auxiliary information for each source. Here, "score" refers to the upstream relevance score and "ascore" refers to the adjusted score used for re-ranking. The user has elected to promote the attribute *noscriptcompat* (for compatibility with the NoScript browser extension [3]) and demote the attributes *haspopup*, *hascookiebanner*, and *hasfixednavbar*, as indicated in the "Policy" table. The "Label sources" table lists the configured sources along with their tier numbers and reputation scores.



Not logged in Talk Contributions Create account Log in

Article Talk Read Edit View history Search Wikipedia

WIKIPEDIA The Free Encyclopedia

Main page
Contents
Current events
Random article
About Wikipedia
Contact us
Donate

Contribute
Help
Learn to edit
Community portal
Recent changes
Upload file

Tools
What links here
Related changes
Special pages

Privacy

From Wikipedia, the free encyclopedia

For other uses, see Privacy (disambiguation).

Privacy (UK: /ˈprɪvəsiː/, US: /ˈpraɪ-/)[1][2] is the ability of an individual or group to seclude themselves or information about themselves, and thereby express themselves selectively.

When something is private to a person, it usually means that something is inherently special or sensitive to them. The domain of privacy partially overlaps with security, which can include the concepts of appropriate use and protection of information. Privacy may also take the form of bodily integrity. The right not to be subjected to unsanctioned invasions of privacy by the government, corporations, or individuals is part of many countries' privacy laws, and in some cases, constitutions.

The concept of universal individual privacy is a modern concept primarily associated with Western culture, particularly British and North American, and remained virtually unknown in some cultures until recent times. Now, most cultures recognize the ability of individuals to withhold certain parts of personal information from wider society. With the rise of technology, the

URL: https://en.wikipedia.org/wiki/Privacy
Label:
Value: True False
Submit

Current labels for https://en.wikipedia.org/wiki/Privacy:

- **hasfixednavbar**:
  - me(0): 0
  - coop(1): 0
  - a(2): 0
  - b(2): 0
- **haspopup**:
  - coop(1): 0
  - a(2): 0
  - b(2): 1
- **noscriptcompat**:
  - a(2): 1
  - b(2): 1
- **hascookiebanner**:
  - a(2): 0
  - b(2): 0

Figure 2: Sidebar for the user to inspect label records for a page and enter their own.

Table 1: Performance of `purerep` on a 2008 2.6GHz Core 2 Duo T9500

Number of labels (thousands)	50	100	500	1000	5000
Time (s)	0.388	0.802	4.524	9.146	43.265
Memory usage (KB)	62301	125077	667206	1292594	6123148

Content-based filtering uses a set of predefined attributes for each item, such as a database of song attributes used to recommend music. Typically the user gives feedback on each item via binary or unidimensional ratings, and an algorithm builds a model of which attributes the user prefers. The PURE approach is almost the opposite: Instead of predicting which attributes a user likes, we predict which attributes an item has.

Collaborative filtering recommends content to a user based on similar users' responses to the same content. Determining similarity between users is almost the same as determining agreement between label sources, and complex algorithms for collaborative filtering could be used in place of our simple algorithm. The main difference is that PURE uses agreement between sources to estimate the applicability of specific attributes, not the overall rating for a resource.

Recommender systems are usually designed to run on the server side, and they expect access to user data that would not typically be available to clients. PURE is distinguished by doing all processing of ratings (labels) on the client side, thus limiting the flow of behavioral data to providers' servers (a foundational goal of data co-ops as conceived by [25, 28]) and maximizing the independence of users in choosing whom to trust. Users are independent in that they need not depend on the servers of any particular company or even depend on a particular data co-op; their client-side tools can process labels supplied by multiple co-ops, and terminating a relationship with one need not disrupt the user experience in a major way.

Finally, recommender systems' goal of maximizing user "engagement" implicitly suggests that engagement *per se* is good and that users can be assumed to act in their own best interests when they give feedback to the system. Much of the popular ire that has recently been directed at online service providers, however, stems from the knowledge that users can be induced to act against their interests, *e.g.*, by scrolling for longer than intended, succumbing to "clickbait" headlines, becoming distracted, or sacrificing privacy for the sake of convenience to a greater extent than they realize and to an extent that they later regret. Certain security practices recognize this reality in acute cases, such as phishing emails, but the ad-supported software industry that dominates much of the data economy has failed to grapple with it effectively. PURE empowers users to make deliberate, informed decisions about their Web use *ahead of time*, rather than making implicit, reactive, often uninformed decisions framed by providers in the form of feedback. This change in perspective, assumptions, and locus of data processing reduces the ability of service providers to induce behaviors that benefit them more than users.

2.5 Performance Measurements and Implications

To achieve the benefits of client-side processing described in Sec. 2.1, it must be feasible to do the necessary processing on a typical client device. We measured the time and memory usage for `purerep` to process a file of randomly generated labels with 8 sources, 8 attributes, and random names ranging from 40 to 80 characters. Measurements for 50 thousand through 5 million labels (averaged over 10 runs) are given in Table 1. `Purerep` is implemented using a trie to achieve linear time complexity at the cost of additional memory usage; as expected, the computation time scales linearly with the number of labels and the memory usage is somewhat high.

Ideally, a user would be able to store and process labels covering all attributes for any resource they might encounter. If a user does 10 searches per day that each return 100 results, it would require up to about 3 million specific labels to cover 8 attributes for all results encountered in a year. This may be an underestimate of what is needed: A user may care about more than 8 attributes, they may want to remember agreements and conflicts between sources for more than a year, and the stored labels won't align perfectly with what happens to turn up in a search.

On the other hand, the number of labels required may be greatly reduced through the use of generic labels. Additional programs may be added to discard or combine certain labels according to heuristics. `Purerep` could also be reimplemented using a different algorithm with different characteristics, for example to use less memory at the cost of more CPU cycles and possibly caching certain intermediate values on disk.

3 VALUE PROPOSITION AND PLAUSIBILITY OF ADOPTION

3.1 PURE and Data Co-ops in Practice: Addressing User-Hostile Design

We now explain how PURE and data co-ops could be used in practice, with a particular focus on user-hostile design as introduced in Sec. 1.3.3.

User-hostile design elements are ubiquitous in everyday computer use: time-delayed paywalls, pop-ups asking for the user's email address, banners demanding consent for cookies that are not necessary for the functioning of a website, *etc.* Brignull's taxonomy [11] of dark patterns catalogues especially blatant varieties of user-hostile design, such as price-comparison prevention, misdirection, hidden costs, bait and switch, confirmshaming (*i.e.*, "guilting" a user into opting into something), and disguised ads.

The presence of these elements is driven by the economics of ad-supported services, and their mechanism of value extraction relies crucially on the user's conscious and unprotected interaction with them. For example, time-delayed paywalls are designed to

wait until the user has become invested in a piece of content before blocking it with a demand for payment. The user must then decide whether to leave or pay for the content from a position clouded by momentary engagement and the sunk-cost fallacy. Such examples present a stark illustration of the implicit-negotiation dynamic described in Sec. 1.4.1, which helps explain the *hositility* toward users whom a service nonetheless relies on. PURE is a natural solution for many user-hostile elements, because it allows the user to avoid the cost of interacting with them altogether, provided they are well labeled. Data co-ops can coordinate this activity on a large scale and negotiate explicitly with service providers to reduce the use of these elements.

3.1.1 *Formation of a data co-op.*

Suppose Alice and Bob share a frustration with pop-ups. They used conventional pop-up blockers in the '90s and '00s and were happy when pop-up blocking became a default feature of Internet Explorer in 2004 [27]. But recently, they have been seeing more and more pop-ups implemented as JavaScript modals, which are not caught by conventional blockers.

They begin using PURESearch to identify web pages with pop-ups and filter them out of their search results. They add each other as label sources in their PURESearch configuration and periodically exchange labels with each other to expand the number of pages each has covered. (The current PURESearch iteration would require each to upload his or her labels to a web server to be accessed at a fixed URL.)

Word gets out about Alice and Bob's pop-up site list, and they coordinate the sharing of labels among more and more users. Soon the ingress of members makes it impractical for everyone to add everyone else manually as a label source and maintain such a long list of sources, so Alice, Bob, and Carol (an enthusiastic early member of the group) decide to set up a server dedicated to the collection, verification, and distribution of PURE labels, and they share it with the group along with a means for users to donate. A data co-op has been born.

3.1.2 *Operational concerns and funding.*

Donation has a lower operational cost than any other funding mechanism, but donations may not be sufficient to support a large number of non-paying users. The donation model also incentivizes the organization to give donors increased say or special treatment, threatening the democratic goal of data co-ops.

Membership dues create a barrier to the use of co-op resources, reducing the number of members, but also ensuring that there is sufficient funding to provide service to the membership. Dues also make the co-op administration maximally accountable to the membership and encourage the membership to participate and provide valuable feedback. Members could also perform tasks for the organization in lieu of dues, such as producing or verifying labels.

A co-op may take a range of approaches to offering exclusive value to their members. If the primary value provided by a co-op is high-quality labels, we believe it is too costly to attempt to prevent members from sharing the labels with non-members. Instead, a co-op can limit the use of its label *servers* by issuing credentials that members' PURE software will use when requesting labels. Joining the co-op could then provide better availability as well as trustworthiness: Obtaining labels directly from a co-op eliminates the middleman who may be re-publishing the same labels faithfully or may be altering them. A co-op could also offer its labels to the general public on a delayed basis, expanding the user base while retaining some benefits for dues-paying members. A co-op may also offer value to its members in the form of tech support or in democratics rights within the co-op.

Co-ops may experiment with sliding-scale dues to accommodate low-income members; truly universal availability of co-op services could be achieved via government funding.

3.1.3 *Label production and verification.*

As discussed in Sec. 2.5, millions of specific labels are required to cover all pages that a user is likely to encounter. Generic labels can reduce the volume of labels necessary, but they also take longer to produce because the labeler must visit enough pages with a given prefix to be confident that the generic assertion for that prefix is correct. If it takes 15 seconds to determine whether a page has a pop-up, it would take over 4000 hours to produce a million specific labels for that attribute. This is clearly more than Alice, Bob, and Carol could do on their own; so, in order for their co-op to provide enough labels to effectively filter out pages with pop-ups, they will have to use alternative methods to collect and verify labels.

The most obvious is to source labels from the co-op membership. PURESearch could be extended to automatically send URLs to the co-op that are not already covered by the co-op's labels, perhaps restricted to results that actually appear to the user or that the user clicks on. Labels for these URLs could be collected passively from members who have such labels on their machines (and have opted in to such collection) or actively by prompting volunteers to check whether certain pages have pop-ups. Rather than verifying each label sourced from the membership, co-op staff could verify a subset of labels from each source to produce PURE reputation and expectation values for the sources and labels, and they could automatically verify a label if the expectation passes a certain threshold. This method could be used for member-generated labels as well as labels from third-party sources, if there are any.

Much of the above computation could be done on each member's client machine by publishing the labels collected by the co-op as well as their own labels representing the work done by staff for verification. However, this limits the resources available for the computation and restricts the algorithm to the one that is running on client machines. It would also present a privacy concern for members who may be willing to share their labels with the co-op staff but would not want them shared with other members.

To determine whether pages have pop-ups, a co-op could also experiment with automated detection, either by processing the rendered page with computer vision or by analyzing the code running on a page and any JavaScript events it produces. Different methods of automated detection could be evaluated internally and subject to various methods of verification, as with other label sources.

Co-op members may want labels for more than just whether a page has pop-ups. An attribute for confirmshaming could be assigned to pages with pop-ups that shame the user when they try to close it (*e.g.* "no, I don't want to make my life easier" instead of "close"). Different attributes could apply to different varieties of pop-ups, possibly distinguishing pop-ups from paywalls and

cookie-consent banners. The co-op could expand to cover other forms of user-hostile design or whatever the membership demands. There are many possibilities, but co-ops should limit the set of attributes as much as possible in order to reduce the overhead of producing and verifying labels and maximize the coverage for the attributes that are supported.

3.2 Creating a Fairer Balance

In this section, we flesh out the reasons we believe that data co-ops and PURE constitute a good value proposition for users and a plausible path to a fairer and more productive balance of power in the data economy.

Ease of use and low overhead: Although many people care about lack of privacy and inability to exploit valuable data that they themselves created, few people care *enough* about this pervasive unfairness to be willing to put thought and effort into combatting it on an ongoing basis. As the description of typical usage in Sec. 2.2 makes clear, no ongoing thought and effort is required for a co-op member to use PURE. Users need not configure their own search interfaces; they can use the default configuration that is provided and updated by the co-op's technical staff. Similarly, users *may* enter their own label assertions, but they are not required to do so.

We expect there to be some users who regularly enter label assertions and, more generally, act consciously to advance the mission of the co-op. A small number of users with technical skills and strong commitment to the cause may join the co-op's technical staff as paid employees. One sees the same range of engagement in other types of membership organizations. For example, most people who work in unionized industries simply join the union, pay their dues, and reap the benefits of collective bargaining; some are more active in the union's negotiations with the employer or in its political or social activities; and a few seek paid employment in official union-leadership positions. Members of professional societies behave similarly: In ACM, for example, most computer-science researchers who join do so for some tangible benefit, such as lower conference-registration fees, that more than compensates for their membership dues; a smaller number play active roles in ACM activities and SIGs; and a very small number run for elected-leadership positions.

An idea whose time has come: As explained in Sec. 1, technical mechanisms like PURE are rarely adopted unless they are enforced legally, incentivized financially, or supported by social trends or norms. In recent years, our society has been trending toward resentment of Big Tech and hunger for protection against Big Tech's predations. Data co-ops can provide protection as well as leverage for users who wish to channel their resentment constructively. Co-ops are also well situated to support socially beneficial norms of online behavior and to provide forums for discussion and evolution of such norms. This point has been made in the popular press (see, *e.g.*, Tarnoff [33] for a recent example) as well as in the technical community.

Broad applicability of PURE: The PURE approach is optimized for flexibility. Labels can apply to any networked resource with a name, and the labels themselves can refer not only to privacy or control of data but to any property that users care about. For example, websites can be labeled according to how well they deal with various forms of harmful content over which there has been recent public concern: false or misleading information that may have contributed to recent political turmoil in many western countries; addictive and manipulative applications and services, which may negatively impact the mental well being of users, especially teens; and misinformation about COVID19 vaccines.

Labels with varying semantic properties could help address each of these issues. A diverse range of data co-ops could produce or curate labels for different domains and purposes. Labels may also be provided by individuals and other types of institutions; of particular interest might be labels that are computed by technologically sophisticated, large-scale "Internet observatories" that measure and analyze phenomena that cannot be observed by a single user or even a typical co-op, which may be relatively homogeneous geographically or demographically. By using a semantically diverse and powerful range of labels, a co-op could, for example, deploy a singular browser extension that marks certain pieces of content as false or misleading. The more ambitious goal of comprehensively optimizing the Internet experience on behalf of co-op members may also be within reach.

In summary, the PURE labeling framework is transparent with respect to the semantics of each label and the objects that labels may apply to. It can serve a very broad range of uses – not only data co-ops as we have conceived them.

Broad applicability of data co-ops: As exemplified by the cases explored in Secs. 1.3 and 3.1, data co-ops can vary widely with respect to size, focus, and technical and social sophistication. Almost any community bound together by a common concern or set of concerns about online life, ranging from the very specific (*e.g.*, local high-school students' inability to access valuable health information online, because it has been inaccurately labeled as pornography) to the very general (*e.g.*, excessive collection of behavioral data by small or mid-sized Internet-based companies, in possible violation of state laws such as the California Consumer Privacy Act) could become better informed and more able to resist by forming a co-op and using appropriate client-side tools.

4 CONCLUSIONS AND FUTURE WORK

4.1 Practical Usability Improvements

The architecture of PURE enables flexible delegation so that users need not be experts, but the current ecosystem of PURE tools is limited to the experimental programs presented here, which require some expertise to use. In order to convince people to use PURE or to join a data co-op that offers PURE services, tools should be developed and refined for improved ease of use and installation. Some domains of PURE usage may warrant specialized tools to validate certain attributes or to evaluate certain classes of content. The goal of usability should be balanced against simplicity of implementation, which is important to maximize accountability and avoid dependence on any particular developer or organization. Methods of software distribution and installation should also be carefully considered with an awareness of the tradeoffs between convenience and centralization of control.

4.2 Possible Extensions of PURE labels

The version of PURE presented here is meant to give a clear illustration of the purpose of PURE labels and what it means for label sources and rated items to be "untrusted." Simplicity is necessary in general to maximize the ability of users or trusted technical experts to understand and control the software they run. However, extensions to the current version of PURE could certainly improve its usefulness.

Different label processing algorithms, possibly adapted from prior work in recommender systems, could be used for different subsets of label records. For instance, the Influence Limiter of Resnick and Sami [31] limits the capacity for manipulation by a malicious rater able to create a bounded number of sybils. This would be well suited to a special lowest tier of label sources that are imported without manual vetting, possibly including individuals who publish their label records to a public registry.

Labels could have a continuum of values rather than binary 0 or 1, which could signify uncertainty on the part of the label source or ambiguity in the applicability of the attribute. Values could also include symbols that specify the semantics of the assertion, describing how the attribute relates to the content rather than just the degree to which it applies. Another variant could include a field identifying a second resource, making it possible to express relations between two resources rather than describing a single resource with a limited vocabulary of attributes and values. These extensions would raise the complexity and resource intensivity of label processing, as well as increase the cost of verification, because complex values may take more time for a human to determine than a simple "yes" or "no."

Labels could also refer to properties that cannot be verified by an end user looking at the content, such as authorship or copyright status, or aggregate ratings of privacy practices. This would require other accountability mechanisms to make the labels trustworthy. For example, lying about the copyright status or origin of a work could incur legal liability. Unverifiable labels could also be retrieved from trusted sources such as a data co-op with strong internal accountability mechanisms.

Finally, label sources could be enabled to make higher-order assertions about other label sources, facilitating delegation, detraction, and third-party vetting. This would warrant careful limits on the types of assertions allowed in order to prevent label records from becoming too difficult to understand.

4.3 Support for Alternative Protocols

We have presented the PURE labeling framework and the PURESearch tool as powerful and flexible ways to promote online privacy and usability and to increase the share of value that users receive from the data that they create. Although the current iteration of PURESearch is designed to improve usage of the existing Web, PURE establishes a basis for much more radical changes to Internet usage. PURE labels are separate from and independent of the resources they refer to; thus, the issue of compatibility between a label record and its object does not arise as long as the object has a name. This shields PURE from the burden of maintaining compatibility with the Web as it rapidly evolves. It also makes PURE well suited to alternative protocols such as Gemini [5] and Gopher [8, 24], which have developed niche followings as a result of discontent with the modern Web and the impossibility of maintaining a modern Web browser without significant capital investment.

Data co-ops could act as incubators for alternative protocols by curating PURE labels and providing client software. The concept of a client-side resource discovery tool, illustrated by PURESearch, suggests software that could display resources using an alternative protocol alongside the more familiar and numerous resources on the existing Web.

We believe such a protocol would be well served by a proposed future Internet architecture called Named Data Networking (NDN) [35]. NDN offers a way to make the Internet more useful and reliable, while making collection of behavioral-data access patterns on a massive scale more difficult. This is due to in-network caching and the lack of source information in data-request packets. A data co-op operating in a geographic locality would be well positioned to operate a set of NDN nodes, eliminating the bulk of external requests for popular content while saving bandwidth and enhancing the user experience.

Immutable named data objects (NDOs), a core component of NDN, also form an attractive basis for labeling, as mentioned in Sec. 2.3. URLs are rough identifiers of content, because the content accessed at a URL can change depending on when it is accessed and the IP address or user agent of the requester. In fact, there is nothing preventing a web server from serving different content for each request. This makes URLs a shaky basis for labels that are meant to refer to a particular piece of content. NDOs do not have this problem, because they consist of immutable data objects cryptographically bound to a canonical name and the publisher's public key. Resources organized as NDOs would thus be much better suited to labeling, and dishonest labelers would no longer have plausible deniability when issuing faulty labels.

In addition to a new data transport protocol, we see potential utility in a new semantic hypertext language designed with PURE and data co-ops in mind, particularly with respect to the goal of user control over client software discussed in Sec. 1.4.2: The behavior of an HTML document running JavaScript is impossible to determine in a bounded amount of time; bounds on the resource usage of scripts can allow for some reasoning about the properties of a page, but the details remain fundamentally opaque. HTML and JavaScript could be replaced by a purely declarative markup language, giving the client program full insight into what a page is doing. Gemini shares this goal but lacks functionalities that users of the modern Web have come to expect. Such functionalities, which are traditionally left to scripts, could be replaced by declarative markup with well defined behavior, giving the user full control over the behavior of the client. Extensions to standard markup could be evaluated and labeled by data co-ops or third parties, with assurances that a markup extension includes no unnecessary data collection or other malicious features.

ACKNOWLEDGMENTS

We thank Kobbi Nissim, Katrina Ligett, Lixia Zhang, and the NSPW participants for their input. This work was supported in part by NSF grant CCF-2131356.

REFERENCES

[1] [n. d.]. *Browser Market Share Worldwide.* Retrieved May 30, 2022 from https://gs.statcounter.com/browser-market-share

[2] [n. d.]. *Edbrowse, a Command Line Editor Browser.* Retrieved May 30, 2022 from http://edbrowse.org/

[3] [n. d.]. *NoScript - JavaScript/Java/Flash blocker for a safer Firefox experience! - what is it? - InformAction.* Retrieved November 17, 2021 from https://noscript.net/

[4] [n. d.]. *Privacy Badger.* Retrieved May 30, 2022 from https://privacybadger.org/

[5] [n. d.]. *Project Gemini.* Retrieved October 27, 2021 from gemini://gemini.circumlunar.space/

[6] [n. d.]. *SCC.* Retrieved May 30, 2022 from http://www.simple-cc.org/

[7] [n. d.]. *Searx - Privacy-respecting metasearch engine.* Retrieved October 27, 2021 from https://searx.me/

[8] Nate Anderson. 2009. The Web may have won, but Gopher tunnels on. *Ars Technica* (4 Nov. 2009). Retrieved October 27, 2021 from https://arstechnica.com/tech-policy/2009/11/the-web-may-have-won-but-gopher-tunnels-on/

[9] Imanol Arrieta-Ibarra, Leonard Goff, Diego Jiménez-Hernández, Jaron Lanier, and E. Glen Weyl. 2018. Should We Treat Data as Labor? Moving beyond "Free". *AEA Papers and Proceedings* 108 (2018), 38–42.

[10] Fabrice Bellard. [n. d.]. *TCC : Tiny C Compiler.* Retrieved May 30, 2022 from https://bellard.org/tcc/

[11] Harry Brignull. [n. d.]. *Types of deceptive design.* Retrieved May 26, 2022 from https://www.deceptive.design/types

[12] Harry Brignull. 2011. Dark Patterns: Deception vs. Honesty in UI Design. *A List Apart* (1 Nov. 2011). Retrieved May 26, 2022 from https://alistapart.com/article/dark-patterns-deception-vs.-honesty-in-ui-design/

[13] Yanghua Chu, Peter Lipp, Philip DesAutels, and Brian LaMacchia. 2009. *PICS Signed Labels (DSig) 1.0 Specification.* https://www.w3.org/TR/REC-DSig-label/

[14] Lorrie Cranor, Brooks Dobbs, Serge Egelman, Giles Hogben, Jack Humphrey, Marc Langheinrich, Massimo Marchiori, Martin Presler-Marshall, Joseph Reagle, Matthias Schunter, David A. Stampley, and Rigo Wenning. 2006. *The Platform for Privacy Preferences 1.1 (P3P1.1) Specification.* https://www.w3.org/TR/P3P11/

[15] Karl Dahlke. [n. d.]. *Command Line Programs for the Blind.* Retrieved May 30, 2022 from http://www.eklhad.net/philosophy.html

[16] Roger Dingledine, Nick Mathewson, and Paul Syverson. 2004. *Tor: The second-generation onion router.* Technical Report. Naval Research Lab Washington DC.

[17] Michael Forney. [n. d.]. *cproc: Small C11 compiler based on QBE.* Retrieved May 30, 2022 from https://sr.ht/~mcf/cproc/

[18] Simson Garfinkel. 1995. *PGP: pretty good privacy.* O'Reilly Media, Inc.

[19] Ayelet Gordon-Tapiero, Alexandra Wood, and Katrina Ligett. 2022. The Case for Establishing a Collective Perspective to Address the Harms of Platform Personalization. In *Proceedings of the 2022 Symposium on Computer Science and Law* (Washington DC, USA) *(CSLAW '22).* Association for Computing Machinery, New York, NY, USA, 119–130. https://doi.org/10.1145/3511265.3550450

[20] Colin M Gray, Yubo Kou, Bryan Battles, Joseph Hoggatt, and Austin L Toombs. 2018. The dark (patterns) side of UX design. In *Proceedings of the 2018 CHI conference on human factors in computing systems (CHI '18).* Association for Computing Machinery, New York, NY, USA, Paper 534. https://doi.org/10.1145/3173574.3174108

[21] Andrei Homescu, Todd Jackson, Stephen Crane, Stefan Brunthaler, Per Larsen, and Michael Franz. 2015. Large-scale automated software diversity—program evolution redux. *IEEE Transactions on Dependable and Secure Computing* 14, 2 (2015), 158–171.

[22] Farnaz Jahanbakhsh, Amy X. Zhang, Adam J. Berinsky, Gordon Pennycook, David G. Rand, and David R. Karger. 2021. Exploring Lightweight Interventions at Posting Time to Reduce the Sharing of Misinformation on Social Media. *Proc. ACM Hum.-Comput. Interact.* 5, CSCW1 (April 2021), Article 18. https://doi.org/10.1145/3449092

[23] Dietmar Jannach, Markus Zanker, Alexander Felfernig, and Gerhard Friedrich. 2010. *Recommender systems: an introduction.* Cambridge University Press.

[24] Cameron Kaiser. [n. d.]. *Why is Gopher Still Relevant?* Retrieved October 27, 2021 from gopher://gopher.floodgap.com/0/gopher/relevance.txt

[25] Katrina Ligett and Kobbi Nissim. 2020. Data Co-ops: Challenges and How to Get There. Video. https://www.youtube.com/watch?v=ZZugFpAOA64 2020 DIMACS Workshop on Co-Development of Computer Science and Law.

[26] Xu Lin, Panagiotis Ilia, and Jason Polakis. 2020. Fill in the blanks: Empirical analysis of the privacy threats of browser form autofill. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security (CCS '20).* Association for Computing Machinery, New York, NY, USA, 507–519. https://doi.org/10.1145/3372297.3417271

[27] Ryan Naraine. 2009. Windows XP SP2 Turns 'On' Pop-up Blocking. *InternetNews.com* (18 March 2009). Retrieved August 10, 2022 from https://web.archive.org/web/20210515082216/http://www.internetnews.com/dev-news/article.php/3327991

[28] Alex Pentland and Thomas Hardjono. 2020. Data Cooperatives. In *Building the New Economy.* https://doi.org/10.21428/ba67f642.0499afe0 https://wip.mitpress.mit.edu/pub/pnxgvubq.

[29] Eric A. Posner and E. Glen Weyl. 2018. *Radical Markets: Uprooting Capitalism and Democracy for a Just Society.* Princeton University Press.

[30] Paul Resnick and James Miller. 1996. PICS: Internet Access Controls without Censorship. *Commun. ACM* 39, 10 (Oct. 1996), 87–93. https://doi.org/10.1145/236156.236175

[31] Paul Resnick and Rahul Sami. 2007. The influence limiter: provably manipulation-resistant recommender systems. In *Proceedings of the 2007 ACM conference on Recommender systems (RecSys '07).* Association for Computing Machinery, New York, NY, USA, 25–32. https://doi.org/10.1145/1297231.1297236

[32] Dave Rupert. 2020. *What is the Value of Browser Diversity?* Retrieved May 30, 2022 from https://daverupert.com/2020/09/the-value-of-browser-diversity/

[33] Ben Tarnoff. 2022. The Internet Is Broken. How Do We Fix It? *The New York Times* (May 27, 2022).

[34] Nicholas Vincent, Hanlin Li, Nicole Tilly, Stevie Chancellor, and Brent Hecht. 2021. Data Leverage: A Framework for Empowering the Public in Its Relationship with Technology Companies. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency* (Virtual Event, Canada) *(FAccT '21).* Association for Computing Machinery, New York, NY, USA, 215–227. https://doi.org/10.1145/3442188.3445885

[35] Lixia Zhang, Alexander Afanasyev, Jeffrey Burke, Van Jacobson, KC Claffy, Patrick Crowley, Christos Papadopoulos, Lan Wang, and Beichuan Zhang. 2014. Named data networking. *ACM SIGCOMM Computer Communication Review* 44, 3 (2014), 66–73.