

Preserving Privacy in Environments with Location-Based Applications

The increase in location-based applications makes protecting personal location information a major challenge. Addressing this challenge requires a mechanism that lets users automate control of their location information, thereby minimizing the extent to which the system intrudes on their lives.

As mobile devices and location systems such as the Global Positioning System (GPS) and phone-based technologies proliferate, so too does interest in location-based applications. These applications include tourist information systems, buddy services that inform users when a friend is nearby, and location-based advertisements, which marketers send to users on the basis of their current locations. Such applications raise serious privacy issues that developers must address, both to appease public concern and to comply with current legislation.

An important first step in protecting users' location privacy is notifying them of requests for this information. For example, a system might ask users to authorize release of their location information by clicking "OK" on a dialog box for each new request. Such a system would be at odds with Mark Weiser's vision of calm technology, however.¹ Weiser argues that for technology to become truly ubiquitous, it should merge into the background such that it becomes a part of the fabric of everyday life. Thus the goal is to minimize technology's intrusiveness and its demands of users. In this article, we address these two conflicting requirements of location-based systems—the

need for users to control their location privacy and the need to minimize the demands made of users.

Our system was motivated by an extremely practical problem—that is, how to protect location information gathered by our groups' location-tracking systems. We created LocServ² to support the various location-based applications developed in our laboratories. LocServ is a middleware service that lies between location-based applications and location-tracking technologies. By unifying location-tracking technologies, LocServ lets location-based applications use multiple positioning systems. In essence, LocServ users can specify a location query using any of the symbolic or geometric location models that LocServ understands, and the service can resolve the queries using any number of underlying technologies. Thus, LocServ allows applications to be written in a way that is entirely independent of the underlying location technology that they use. Such a service requires mechanisms for controlling access to users' location information without repeated user intervention. We have thus developed an extensible system that gives users fine-grained control over the release of their location information. More specifically, we offer a general framework of components that lets users apply general policies to control distribution of their information. We use factors such as the type of organization or application requesting the data together

Ginger Myles
University of Arizona

Adrian Friday
Lancaster University

Nigel Davies
University of Arizona and
Lancaster University

with its information retention and distribution policies and, crucially, a mechanism for consulting external entities such as application-specific modules before releasing information.

Like the World Wide Web Consortium's Platform for Privacy Preferences (P3P) and Marc Langheinrich's Privacy Awareness System (pawS), our system uses machine-readable privacy policies and user preferences to automate the privacy management decision-making process. As the sidebar, "Related Work on Privacy in Location-Based Systems" explains, however, our system architecture and preference language are significantly different from those in P3P and pawS, reflecting the differences in deployment domain.

General requirements

The following scenario illustrates the requirements for a privacy policy system for location-based applications.

A health-care worker, Sally, carries a mobile phone that lets her employer locate her whenever she has the device. During the day, Sally visits patients in their homes and is pleased that her company can locate her with an accuracy of about 100 meters. Her company uses this information to inform patients of her likely arrival times and to maintain her schedule without her having to disclose her movements within houses.

When she is off-duty, Sally does not want the company to track her. If she must wait for a table at a restaurant, however, she wants the restaurant to know her location so that they can find her when her table is ready. Of course once she leaves the restaurant, she does not want the owners to be able to determine her location. Similarly, when Sally uses her mobile phone to call a cab at the end of the evening, she wants the taxi company to determine her location automatically to ensure a smooth pickup, but she does not want them to be able to trace her once the journey is over.

Sally is also planning a vacation, and when she visits, for example, the FunTime amusement park, she will let the park management track her for safety and management purposes, but she does not want this information correlated with her identity.

Finally, Sally uses a small set of location-based applications, including a general information service

that warns her of traffic holdups in her area and a find-a-friend service that tells her when she is within half a mile of one of her friends. Independent companies provide these services, and Sally wants to disclose only the minimum amount of information necessary for them to provide the required functionality.

This scenario also illustrates the richness of constraints (or preferences) users might want to apply to control the distribution of their location information. In our example, Sally restricts access to her information in several ways:

- **Organization.** In most cases, Sally restricts access to her location information to specific organizations (such as her employer or the find-a-friend service provider).
- **Service.** While Sally generally restricts access to her location information to known companies, she will also accept certain types of information (information bulletins, for example) from new companies.
- **Time.** Sally controls her employer's access to her location information on the basis of the time of day—she has different policies for work days, weekends, and evenings.
- **Location.** Sally will let the amusement park management track her location while she is in the park but not when she leaves. Hence, their ability to obtain location information depends on her location and the relationship between the company requesting the information and the physical space.
- **Request type.** Sally restricts the type of query an application can issue to obtain her location information. For example, she provides an anonymized trace of her location in the amusement park. Furthermore, although Sally will tell the find-a-friend service when she is within

a specified distance of one of her friends, she does not want to tell them her exact location.

- **Context.** Sally uses her calendar (or work schedule) to control her employer's access to her location information. Other forms of contextual information, such as whether or not she is alone, could easily form part of her preferences.

This is not an exhaustive set of requirements for privacy preferences in ubiquitous computing environments. Indeed, a user could easily add the constraint, "infor-

Our system uses machine-readable privacy policies and user preferences to automate the privacy management decision-making process.

mation about my child's location can only be released when the child is with me or my spouse." At least two additional constraints exist: the need to comply with existing and emerging legislation and, as discussed previously, the desire to minimize user interaction when dealing with requests for location information.

Current legislation

Currently, two important pieces of privacy-related legislation exist: the US Privacy Act of 1974 and the European Union's Directive 95/46/EC.³ The US Privacy Act of 1974 was designed for information privacy. It gives legal substance to the idea of fair information practices including openness and transparency (for example, no secret record keeping), individual participation, collection and use limitations, reasonable security, and accountability.

The EU's directive addresses the protection and movement of personal data. It limits data transfer to non-EU countries to those countries deemed to have an adequate level of privacy protection. It also requires explicit consent—that is, a user must unambiguously consent to having their information collected. Like Langheinrich, we recognize the difficulty of design-

Related Work on Privacy in Location-Based Systems

Despite its obvious importance, relatively little systems-oriented research has addressed privacy protection in ubiquitous computing systems. Several exceptions have informed our work.

Geopriv

The Internet Engineering Task Force's Geopriv working group has identified a need to "securely gather and transfer location information for location services, while at the same time protecting the privacy of the individuals involved."¹ The November 2002 IETF draft describes one approach to securely transferring location information and associated privacy data. In essence, the scheme involves creating *location objects* that encapsulate user location data and associated privacy requirements. Central to this scheme is the notion that location objects can be made tamper resistant—for example, by digitally signing them. The approach is similar to digital rights management schemes designed to protect digital media from illegal redistribution.

At present the Geopriv proposal and our scheme are largely orthogonal. We focus on defining the management of information release and the nature of the privacy rules and preferences used to control this release, but Geopriv does not address these issues. Hence, Geopriv could use a subset of our system's rule sets and overall policy architecture as the "privacy-enabling information"¹ stored in a location object. Geopriv's proposed coupling of data and privacy metadata offers greater accountability than our system when location information has been passed between multiple applications. The practicality of such a system has yet to be determined, however.

P3P and Appel

Together, P3P² and Appel³ help Web sites announce their privacy practices while letting users automate their accept and reject decisions. P3P specifies an architecture comprising user agents, privacy reference files, and privacy policies. When users access a Web site, their user agents obtain a privacy reference file for the Web site using one of several well-defined mechanisms. This file contains a list of mappings between URIs for the site's Web resources and URIs of their associated privacy policies. The Web agent can thus ensure that the system downloads, parses, and compares the appropriate privacy policy with the user's preferences prior to accessing a Web resource.

P3P also specifies the language used to express privacy policies, while privacy preferences used to configure user agents can be expressed in several forms. P3P commonly uses Appel to ensure that different user agents can reuse preferences.

P3P does not attempt to enforce or ensure privacy through technology—for example, by cryptographic or anonymization techniques. Instead it relies on social and legal pressures to compel organizations to comply with their stated policies.

pawS

pawS is a privacy awareness system for ubiquitous computing environments.⁴ Like P3P, *pawS* aims to provide users with tools that let them protect their personal privacy and help others respect that privacy. It is based on respect and social and legal norms rather than rigorous technical protection of private information. In *pawS*, when a user enters an environment in which services are collecting data, a privacy beacon announces the privacy policies of each service in the

ing a system that guarantees compliance with such laws and hence developed an architecture based on trust and respect: producers and consumers both state their information collection and distribution policies, and we assume these statements are accurate. We rely on digital signatures to prove the statements' authenticity and expect that legislation could be used to hold users accountable for violating their stated policies.

User interaction

We aimed to develop a system that required minimal ongoing user involvement. In particular, we did not want users to have to repeatedly evaluate the acceptability of an application's request for location information. Instead, we wanted to

push a request's acceptance or rejection to the periphery and only bring a request to users' attention if they had not established a policy to handle it. The potentially large volume of requests each user could be subjected to on any given day made this an important design consideration. Moreover, we believe user privacy should be protected by default; thus, the system architecture lets a user elect to share certain information rather than protect specific information.

System architecture

Figure 1 shows our system's overall architecture. We assume the existence of a location server (such as LocServ) that answers applications' queries concerning users' locations. These queries can be broadly classified into types:

- *User location requests*—requests for the location of a specific user or users, identified by their unique identifiers.
- *Enumeration requests*—requests for lists of users at specific locations, expressed either in terms of geographic or symbolic attributes.
- *Asynchronous requests*—requests for "event" information, such as when users enter or leave specific areas or when proximity relationships are satisfied (for example, tell me when Sally and Bob are within half a mile of each other).

The location server abstracts over the underlying positioning technologies used to derive location, such as GPS and Active Bat systems,⁴ and provides applications with a common API. Users subscribe to

environment. A user's privacy proxy (similar to P3P's user agents) checks these policies, expressed in the same language as P3P policies, against the user's predefined privacy preferences, expressed in Appel. If the policies agree, the services can collect information and users can utilize the services. If the policies don't agree, the system notifies the user, who can choose not to use the service in question or, in extreme cases, leave the area in which the information collection is occurring.

Systems analysis

P3P and pawS are good starting points for investigations into privacy-enabling schemes in ubiquitous computing, and significant work has gone into making P3P comply with existing and emerging legislation in information protection and privacy. However, neither system can adequately support the scenarios described in the article. Because P3P is designed to support Web interactions typically involving e-commerce and business applications, its mechanisms for obtaining reference files and policy documents are tightly coupled with Web usage models, protocols, and deployment architectures. Moreover, the policy language, while extensible, contains constructs for expressing information-collection and management policies appropriate for protecting information disclosed during Web browsing and user-initiated online transactions. Our system protects the user when arbitrary third-party location-based applications require information from the user's location server. Similarly, while Appel provides a good starting point for expressing privacy preferences, it cannot support the richness of expression necessary for autonomous evaluation of user criteria in real application domains.

Our system also differs from pawS, P3P, and Geopriv in its association between preferences, services, and data. Geopriv's fairly simple model for associating privacy requirements with user data makes it difficult to capture privacy requirements that span multiple data items or do not readily fit into location objects. For example, in Geopriv it would be difficult to specify that an application can know if two people are in the same general area but cannot know their individual locations.

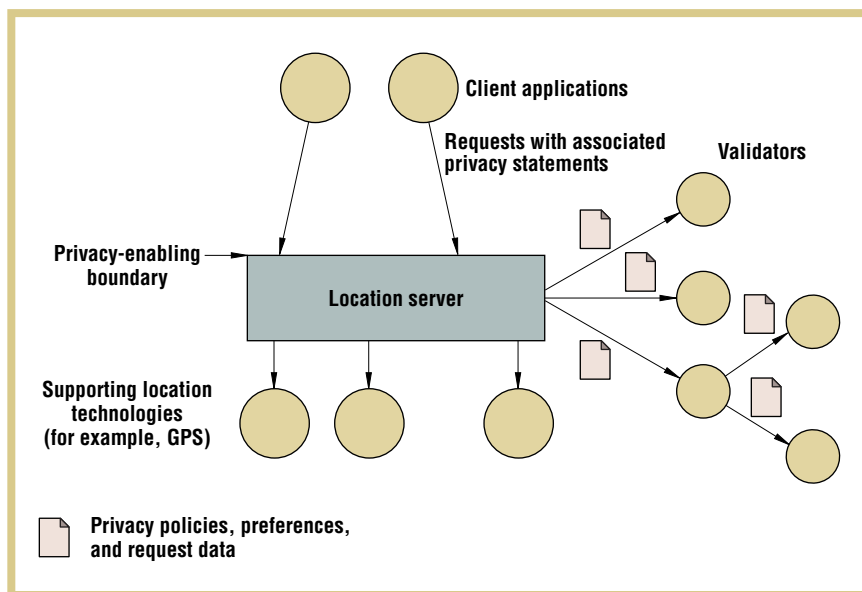
Philosophically, our system is fundamentally different from pawS. pawS lets users protect their privacy at the moment of information capture, typically when they access a service or enter a new geographic space. In contrast, our system attempts to provide privacy checks at the moment of information release—that is, when an application makes a solicited or unsolicited request for location information.

REFERENCES

1. J. Cuellar, J.B. Morris Jr., and D. Mulligan, "Geopriv Requirements," Internet draft, Nov. 2002.
2. *The Platform for Privacy Preferences 1.0 (P3P1.0) Specification*, World Wide Web Consortium, Sept. 2001, www.w3.org/TR/2001/WD-P3P-20010928.
3. *A P3P Preference Exchange Language 1.0 (Appel 1.0)*, working draft, World Wide Web Consortium, Apr. 2002, www.w3.org/TR/P3P-preferences.
4. M. Langheinrich, "A Privacy Awareness System for Ubiquitous Computing Environments," *Proc. Ubicomp*, LNCS 2498, Springer-Verlag, 2002, pp. 237–245.

one or more location servers, registering their privacy requirements with each server. These requirements take the form of system components called *validators*. Given a request for location information and a privacy policy provided by the application, validators determine whether the requested information can be released and, if so, whether the location server should impose

Figure 1. Overall system architecture. The location server works on top of positioning technologies such as GPS, receiving requests for users' location information from various applications. System components called *validators* determine whether or not to grant the applications' requests.



any special constraints (such as reducing the location data's accuracy). Users register a single validator with a location server for each of their identifiers. Because validators can call other validators for help making decisions, however, multiple validators might be used to determine the correct response in any given case. We assume that trusted relationships exist between users, their location-tracking systems, validators, and the location server.

Applications wishing to obtain information about a user's location query the location server, including with the query a

the location server and sends both a query and a statement of its privacy policy. The query should be in a format the location server in question understands. In our system, this is a proprietary LocServ format, but applications could use other formats such as that proposed by the Location Interoperability Forum (www.locationforum.org) for other types of location servers.

To support automatic checking of privacy policies, developers must agree on a common scheme for describing these policies. We use Appel, the policy specification language proposed as part of the W3C's

lets the system introduce certification schemes (a user might wish to only use services certified by a particular set of trusted organizations or authorities).

Purpose. P3P's **purpose** tag reflects its orientation to e-commerce and Web interactions (representing such intentions as telemarketing and Web page tailoring). We use a new set of broad classifications to more accurately reflect the intentions of location-based service providers: safety, entertainment, marketing, information, service delivery, statistical analysis, and security. Safety services include danger warnings in particular geographic areas or emergency service support, whereas security applications track the user for security purposes (surveillance, for example). Like P3P, our schema allows for arbitrary user extensions.

In contrast to P3P, our system does not require policies to specify the data to be collected because the system can determine it from the associated query.

privacy policy statement. The location server consults the relevant validators before releasing any information. If necessary, the location server can require applications to sign their privacy statements and, similarly, applications can require the location server to return a signed agreement to these practices.

Both the location server and the validators are abstract entities that can be realized as integrated components of a single location technology or, as in our system, as a self-contained middleware service with associated internal and external validators balancing efficiency with extensibility. Thus the location server in our architecture simply represents the point at which validators check privacy statements from client applications against users' or system administrators' privacy preferences.

Expressing privacy policies

Third-party applications seeking a user's location first choose a location server responsible for the user, typically determined by the contact details available to the application (given a phone number, for example, an application might contact the location server of the user's mobile telephone provider). The application contacts

P3P specification (see the sidebar). This XML-based language provides a machine-readable form of the privacy policies currently found on many Web sites. It provides a wide range of constructs allowing, for example, Web sites to describe the information types they will collect, who will have access to this information, and how long it will be retained.

To address the requirements of location-based applications, we extended the P3P policy specification language. These extensions constitute the main differences between privacy policies related to Web browsing and e-commerce and those related to location-based applications.

Entity. The P3P **entity** tag provides a mechanism for describing the business and contact details of an organization providing Web-based services. In our system, entities represent arbitrary third-party applications requesting location information. We have augmented the **entity** tag with the fields **type** and **cert** (optional). An organization type can be nonprofit, profit, or government. The tag lets the system automatically identify and filter course-grained application classes (for example, it can place restrictions on all nonprofit services). The **cert** field

Request-initiation. In P3P-based Web interactions, the user always initiates a dialogue by visiting a particular Web site and following certain hyperlinks. In our model applications can request information from the user's location service at any time. We added a new **request initiation** tag and partitioned interactions into two classes: unsolicited and solicited. Unsolicited interactions are not explicitly or consciously triggered by the user (for example, initiated speculatively as a user wanders into a particular region or proximity). We assume solicited interactions have been explicitly triggered by some out-of-band user-initiated action (requesting a taxi to his or her current location, for example). Arbitrary user data (*proof*) might accompany a **solicited** tag to link the request back to the instigating action. The **request initiation** tag also lets the user block unwanted requests from unsolicited services.

In contrast to P3P, our system does not require policies to specify the data to be collected, because the system can determine it from the associated query. Because our policy language is simply a set of extensions to the current P3P language, our system can use existing P3P policy specifications. Such policies are unlikely to be applicable, however, because they will typically specify data items collected during Web-based activities

(and will thus be inappropriate for queries issued to a location service). In addition, they will not detail the application purpose sufficiently to let most users determine whether to accept or reject the request.

Validators and user preferences

Validators check the acceptability of privacy policies and determine whether the system should accept a request. As part of this decision-making process, validators can call other validators, creating networks of components that collectively determine whether information should be released to the application. Potential validator components include

- *User confirmation.* A simple validator component could pass the responsibility for decision making to the user by displaying a dialog box containing a summary of the requesting application's privacy policy and information requirements. While such a validator does not begin to address the desire for calm technology, it is a useful component at the end of a chain of other validators when the system has been unable to automatically decide whether to accept a request.
- *User data and context.* We can construct validators to base their decisions on data from user applications such as calendars or system components such as activity monitors. More generally, validators let the system include contextual information in the decision-making process.
- *External services.* External validation services can, for example, resolve issues of ownership of a physical space, which may be a factor in determining a request's acceptability. Other examples of external services include verification of a third party's reputation and checking "spam request" listings.

A requirement for general-purpose validators that can make decisions based on information supplied in a request's privacy policy also exists. Users could tailor these general-purpose validators using a range of mechanisms including preference languages such as Appel or more general-purpose rule-based languages. Our experiments suggest

that a simple scheme that allows constraints expressed in terms of the basic attributes specifiable in privacy policies (entity, purpose, and so on) provides sufficient flexibility for expressing many common privacy preferences. Additional features include

- *Statement.* The user can define a policy statement for each request type supported by the location server API, choosing to accept the request unconditionally or to impose certain time, location, or accuracy constraints. For example, a user might accept a request to enumerate all users within a given locale (providing course-grained identification), but might reject a direct request for his or her precise location.
- *Limit time.* Users can associate time bounds with preferences. For example, users might impose a constraint such that their employers can only access their location during work hours.

- *Limit location.* Users can restrict collection of their information to specified geographic areas. For example, users might let a shopping mall track their location while they are in the mall, but would presumably want this surveillance to stop when they leave.
- *Validator.* Users can specify one or more URIs to external policy validators. Each validator receives a copy of the third-party request and the accompanying privacy policy statement. The validator can implement any arbitrary policy on the user's behalf, returning accept or reject responses to the location server evaluating the policy. Currently, our system allows validators to be combined using simple logical operators.
- *Quality of service.* Our system incorporates a placeholder for specifying quality of service (QoS) in user preferences,

which can limit the accuracy or certainty with which a user can be located.

Provisioning time and location constraints is particularly important in supporting ongoing location operations that yield events to a third party over time. Users can register any component that provides an appropriate interface as a validator. Thus we do not expect just one validator type or even one mechanism for specifying preferences to general-purpose validators. Hence, we are not concerned with standardizing languages for expressing preferences, such as Appel.

Anonymity

Producing anonymous location information is a nontrivial task, and several recent research papers have attempted to address this problem (see the related article, "Location Privacy in Pervasive Computing," in this issue). Our current system relies on users

Provisioning time and location constraints is particularly important in supporting ongoing location operations that yield events to a third party over time.

having multiple identifiers. More specifically, when the system receives a location request that requires it to divulge a user's identity, the user's validators are consulted to determine whether to return the user's long-term identifier, a short-term identifier associated with the user, or a new randomly generated identifier. The system can create the new identifiers with new validators and associated rule sets, or the new identifiers can inherit the user's original configuration.

Say FunTime's owners have registered with Sally's location service to receive an event whenever someone enters the park. Because Sally is willing to provide them with that information, but not with her identity, her validator sends FunTime a new identifier in response to its query. Of course, FunTime can reuse the identifier to obtain Sally's location during her visit, but the identifier will eventually expire and will have no link

TABLE 1
Policy rule base for a general-purpose validator describing Sally's preferences.

Parameter	Rule 1	Rule 2	Rule 3	Rule 4	Rule 5	Rule 6
Company	MyEmployer.com	MyEmployer.com	*	Taxi.com	*	FindaFriend.com
Organization type	Commercial	Commercial	*	Commercial	Nonprofit, government	Commercial
Certification	*	*	*	*	*	*
Request type	*	*	Enumerate	*	Location, asynchronous	Colocation
Purpose	Safety, information, service delivery, statistics, security, other	Safety, information, service delivery, statistics, security, other	Safety, information, service delivery, security	Service delivery	Information	Service delivery
Retention	*	*	Stated purpose	Stated purpose	Stated purpose	Stated purpose
Distribution	Ours	Ours	Ours	Ours	Ours	Ours
Initiated	*	*	*	Yes	*	*
Validators	None	References a validator that can check Sally's calendar	References a verification service that checks ownership of physical locations	None	None	*
Location	*	*	*	*	*	*
Time	M–F, 9 a.m.–5 p.m.	*	*	*	*	*
Anonymity	None	None	Returns a new pseudo-identifier	None	None	None

* Any

back to Sally's long-term identifier.

Such a scheme does not, of course, provide complete anonymity because applications might be able to deduce the mapping between temporary and permanent identifiers by observing movement patterns. Protecting against this type of attack is outside the scope of this work.

The user's role

Validators automate the process of checking privacy policies against user preferences, whether or not these preferences are an integral part of the validator or passed to the validator as a parameter in the form of a configuration script, for example. Clearly we do not expect end users to write their own validator components or even produce configuration scripts unaided. Instead, we assume that service providers and other trusted organizations will provide users with default validators. For situations requiring nonstandard preferences, simple tools could help users create appropriate configurations. To help clarify this process we created several "wizards" that ask users a series of questions about their location information

privacy preferences and then generate appropriate configuration scripts.

Similarly, we would not expect individual application writers to author their own privacy policies. Rather, we expect companies to include them in their general policy-making process as they do human-readable Web privacy statements. Developers of systems such as P3P, which attempt to provide similar support for privacy management, share this view.

Controlling access to user location information: A demonstration

To illustrate how our scheme can realize our original scenario, we divide the scenario into six separate interactions.

- Sally's employer accesses her location during office hours.
- Sally's employer accesses her location outside of office hours but while she is on call.
- Sally visits a restaurant and lets them locate her.
- Sally calls a taxi and the taxi company

obtains her location information.

- Sally visits the FunTime amusement park.
- Sally uses a find-a-friend service and local information providers.

Validators and preferences

Sally's simple configuration consists of three validators. The first is a general-purpose validator that uses a configuration script to determine whether to accept or decline a request, or to require additional validators for the decision-making process. This validator calls the remaining two validators—one to access Sally's calendar and the other to determine ownership of physical spaces—as required. Table 1 gives a rule base for a general-purpose validator that would meet Sally's requirements. Columns correspond to rules the system uses to allow access to her location information, and rows correspond to either parameters to be matched in the requesting application's privacy policy or statements of action to be taken—for example, to consult an external validator. Several rows relate to parameters found in conventional P3P policies as detailed in the P3P specification.

Figure 2. Sally's employer's statement of privacy policy, illustrating indefinite retention of unsolicited information access.

Rules 1 and 2 control Sally's employer's access to her location information. Rule 1 allows the company to obtain any information it wants on Sally's whereabouts during office hours. Rule 2 supplements Rule 1 by ensuring that when Rule 1 rejects a request, the system will use an additional validator to check Sally's calendar to see if she is on call and hence whether to accept the request.

Rule 3 lets both FunTime and the restaurant access Sally's location. This rule simply states that Sally's location can be determined by anyone who owns the physical space in which Sally is located. However, this information is presented in the form of a pseudo-identifier, thus partially concealing Sally's true identity.

Rule 4 lets the taxi company get Sally's location information when she requests service. The rule specifies that the taxi company can only obtain her location information if she explicitly requests the service. Some form of information exchange would be required to ensure that this condition is met. Rules 5 and 6 restrict access to Sally's location on the basis of company and service type and company name and query type, respectively.

Application privacy policies

Each application that requests Sally's location information must also present a statement of its privacy policy for validators to check. Figure 2 shows an example privacy policy for Sally's employer, and Figure 3 shows a policy for a local travel information service Sally uses. Both policies

Figure 3. Sample policy for a local information service that operates on a nonprofit basis and does not retain any of the location information it obtains.

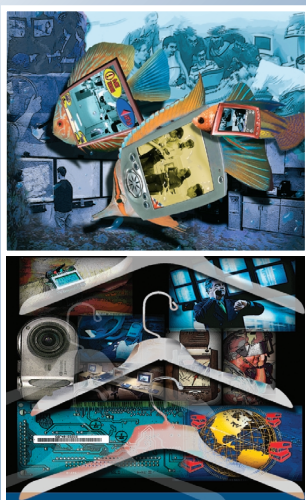
```
<POLICY name="Employer">
  <ENTITY>
    <DATA-GROUP>
      <DATA ref="#business.name">SallysEmployer.com</DATA>
      <DATA ref="#business.address">1 The Grindstone</DATA>
    </DATA-GROUP>
    <TYPE><profit></TYPE>
    <CERT><certified companies 6790></CERT>
  </ENTITY>
  <DISPUTES-GROUP>
    <DISPUTES resolution-type="independent"
      service="http://www.SallysEmployer.com"
      verification="SallysEmployer.com"
      short-description="For disputes please contact the management.">
    <REMEDIES><correct/></REMEDIES>
  </DISPUTES-GROUP>
  <STATEMENT>
    <CONSEQUENCE><consequence.data></CONSEQUENCE>
    <PURPOSE><other-purpose></PURPOSE>
    <RECIPIENT><ours></RECIPIENT>
    <RETENTION><indefinitely></RETENTION>
    <REQUEST-INITIATION><unsolicited></REQUEST-INITIATION>
  </STATEMENT>
</POLICY>
```

```
<POLICY name="Employer">
  <ENTITY>
    <DATA-GROUP>
      <DATA ref="#business.name">Tucson Happy-Travel Service</DATA>
      <DATA ref="#business.address">Tucson, AZ</DATA>
    </DATA-GROUP>
    <TYPE><nonprofit></TYPE>
    <CERT><certified companies 6380></CERT>
  </ENTITY>
  <DISPUTES-GROUP>
    <DISPUTES resolution-type="independent"
      service="http://www.thts.com"
      verification="thts.com"
      short-description="For disputes please contact the national travel service arbitration scheme.">
    <REMEDIES><correct/></REMEDIES>
  </DISPUTES-GROUP>
  <STATEMENT>
    <CONSEQUENCE><consequence.data></CONSEQUENCE>
    <PURPOSE><info></PURPOSE>
    <RECIPIENT><ours></RECIPIENT>
    <RETENTION><no-retention></RETENTION>
    <REQUEST-INITIATION><unsolicited></REQUEST-INITIATION>
  </STATEMENT>
</POLICY>
```




IEEE Pervasive Computing delivers the latest peer-reviewed developments in pervasive, mobile, and ubiquitous computing and acts as a catalyst for realizing the vision of pervasive (or ubiquitous) computing, described by Mark Weiser nearly a decade ago. If you haven't renewed your subscription, visit <http://computer.org/subscribe> or contact our Customer Service department:

+1 800 272 6657
toll-free in the US and Canada
+1 714 821 8380 phone
+1 714 821 4641 fax



contain information specified using the basic P3P vocabulary and our extensions. Policies for the other applications take similar form.

We are currently implementing the system described in this article as part of our ongoing research into technologies to help create deployable pervasive systems—that is, systems that can be deployed outside the confines of the laboratory. We will use this implementation to support two application-oriented projects. The first, a series of extensions to the Lancaster Guide tourist system,⁵ is designed to allow individual users to create their own content for Guide. To explore user reaction to the tension between sharing their information and protecting their privacy, we will allow users to choose from a series of privacy options. The second application-oriented project, a system for pervasive health care based on mobile devices and Grid technologies, will use the privacy system to help reassure patients of the privacy of their medical data. In both cases, we are interested in how users react to privacy issues and, in particular, whether our system allows users to restrict the dissemination of their information according to their wishes. **E**

REFERENCES

1. M. Weiser and J.S. Brown, *Designing Calm Technology*, white paper, Dec. 1995; <http://nano.xerox.com/weiser/calmtech/calmtech.htm>.
2. A. Jacobson et al., "LocServ—A Unifying Location Service," submitted for publication, 2002.
3. M. Langheinrich, "Privacy by Design—Principles of Privacy-Aware Ubiquitous Systems," *Proc. Ubicomp*, LCNS 2201, Springer-Verlag, 2001, pp. 273–291.
4. A. Harter et al., "The Anatomy of a Context-Aware Application," *Proc. 5th Ann. ACM/IEEE Int'l Conf. Mobile Computing and Networking (Mobicom)*, ACM Press, 1999, pp. 59–68.
5. K. Cheverst et al., "Experiences of Developing and Deploying a Context-Aware Tourist Guide: The Lancaster Guide Project," *Proc. 6th Ann. ACM/IEEE Int'l Conf. Mobile Computing and Networking (Mobicom)*, ACM Press, 2000, pp. 20–31.

For more information on this or any other computing topic, please visit our Digital Library at <http://computer.org/publications/dlib>.

the AUTHORS



Ginger Myles is a PhD student in the Computer Science Department at the University of Arizona. Her research interests include privacy and security in ubiquitous computing and software protection. She received her BA in mathematics from Beloit College in Wisconsin. Contact her at the Dept. of Computer Science, Univ. of Arizona, Gould-Simpson 721, PO Box 210077, Tucson, AZ 85721-0077; mylesg@cs.arizona.edu.



Adrian Friday is a lecturer in the Department of Computer Science at Lancaster University, UK, and an active member of the Distributed Multimedia Research Group. His research interests include distributed-system support for mobile, context-sensitive, and ubiquitous computing. He received a BSc from the University of London and a PhD from Lancaster University, both in computer science. He is a member of the ACM, the IEEE, and the BCS. Contact him at the Computing Dept., Faculty of Applied Sciences, Lancaster Univ., Bailrigg, Lancashire LA1 4YR, UK; adrian@comp.lancs.ac.uk.



Nigel Davies is a professor of computer science at Lancaster University and an associate professor of computer science at the University of Arizona. His research interests include systems support for mobile and pervasive computing. Davies focuses in particular on creating deployable systems that can be evaluated using end users. Davies holds a BSc and a PhD in computer science, both from Lancaster University. He is an associate editor of *IEEE Transactions on Mobile Computing and IEEE Pervasive Computing*. Contact him at Computing Dept., Faculty of Applied Sciences, Lancaster Univ., Bailrigg, Lancashire LA1 4YR, UK; nigel@comp.lancs.ac.uk.