APPROACHES FOR MANAGING CONTEXT, INCENTIVIZATION, AND PRIVACY FOR MOBILE-DRIVEN ENVIRONMENTAL AND HEALTH MONITORING APPLICATIONS

by

Michael Daniel Scott

(Under the Direction of Lakshmish Ramaswamy)

Abstract

Effective management of interaction context, user incentivization, and optional privacy controls benefit participatory sensing mobile applications for environmental and health monitoring by improving the quality and quantity of data shared. This work provides a conceptual framework for these challenges and the tradeoffs between: context and resource consumption, context and user effort, privacy and utility, and user incentivization. The expression of these issues is explored through two case study applications–CyanoTracker and PillSafe.

Mobile applications for citizen science may automatically consider and capture the context surrounding interaction with citizen science projects. This data may include the time, date, location, weather conditions, and other data useful to the application.

Encouraging participation in these projects requires consideration of user motivations as well as the potential incentivization mechanisms. These reward systems must value a report in the context of the application as it relates to the quantity and quality of data received. The burden associated with report submission should evaluated and factored in to the incentivization strategies.

Privacy-conscious individuals may be hesitant to participate in citizen science projects which do not provide adequate control over the degree of information sharing. A balance should be struck between preserving privacy and providing high-quality data to the project.

Index words: citizen science, participatory sensing, incentivization, gamification, cyanobacteria, privacy, context

# APPROACHES FOR MANAGING CONTEXT, INCENTIVIZATION, AND PRIVACY FOR MOBILE-DRIVEN ENVIRONMENTAL AND HEALTH MONITORING APPLICATIONS

by

Michael Daniel Scott

B.S., University of Georgia, 2010

A Dissertation Submitted to the Graduate Faculty of The University of Georgia in Partial Fulfillment of the Requirements for the Degree

Doctor of Philosophy

Athens, Georgia

2017

APPROACHES FOR MANAGING CONTEXT, INCENTIVIZATION, AND PRIVACY FOR MOBILE-DRIVEN ENVIRONMENTAL AND HEALTH MONITORING APPLICATIONS

by

Michael Daniel Scott

Major Professor: Lakshmish Ramaswamy

Committee: Hamid Arabnia
Suchi Bhandarkar

Electronic Version Approved:

Suzanne Barbour
Dean of the Graduate School
The University of Georgia
December 2017

# Dedication

This work is dedicated to my parents—Deborah and Michael—and my brother Josh. Without your love and support over the years none of this would have been possible.

## Acknowledgements

I would like to thank my major advisor, Dr. Lakshmish Ramaswamy, for his support and guidance during my time in graduate school. I would also like to thank my entire committee for their time and input to the project.

This research has been partially funded by the National Science Foundation under Grant Number CCF-1442672. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors, and do not necessarily reflect the views of the NSF.

## Table of Contents

## List of Figures

# List of Tables

# Chapter 1

## Introduction

Citizen science provides opportunities to involve the layperson in the scientific process. This allows large-scale participation in monitoring and data collection projects. These contribute to the effectiveness of environmental and health-monitoring programs by increasing the volume of contributed data. The projects that benefit from citizen science approaches range from conservation biology, water quality monitoring, population ecology, and other monitoring programs. It has been noted that citizen science is required for large-scale environmental science [48]. The authors in [6] note that citizen science projects have been remarkably effective in advancing scientific knowledge. It is noted that data may be collected on a larger scale and over longer time periods using volunteer contributions [14].

Mobile and smart devices have enhanced citizen science projects by providing readily available platforms running applications tailored to the individual program needs. These internet-connected devices simplify the communication of collected data with interested parties. The software programs running on these devices face several issues and challenges, including accessibility, changing technology [48], management of interaction context, user incentivization, and privacy protection.

Effective management of interaction context, user incentivization, and optional privacy controls benefit participatory sensing mobile applications for environmental and health monitoring by improving the quality and quantity of data shared. This work examines these three issues in detail as they relate to the implementation of two case-study applications: CyanoTracker and PillSafe. The capture (and use) of contextual information surrounding observations may improve the quality of reports received and may simplify the reporting process. User incentivization concerns convincing users to participate in a project and goes further to help promote long-term engagement. User privacy controls ensure more privacy-conscious users' concerns regarding participation are minimized.

## 1.1 Contributions

The contributions of this thesis are: to provide a conceptual framework for the tradeoffs between context and resource consumption, the capture of context and user effort, the tradeoff between privacy and utility, and user incentivization. Adequately addressing these challenges enable the creation of more user-friendly and effective citizen-science applications.

The capture and use of interaction context reduces the burden placed on the user and may improve the quality of data received. When a user engages in a participatory sensing project they volunteer some of their time and other resources to communicate some sensory data to concerned parties. Deployed on sensor-rich mobile devices, these applications may combine the core observation of the report with any relevant metadata obtained regarding the incident. These report details may also be synthesized with external data sources (e.g. weather, satellite imagery) upon report filing or may be done afterwards. When a user self-reports a geographic location they may make mistakes or may only know the nearest street address. This user-provided location description may not be precise enough for some applications. The use of GPS coordinates obtained by the device provides a more reliable source of information. By leveraging available information sources we simplify the interaction with citizen-science applications and improve the quality of transmitted data.

User incentivization is a particular challenge facing citizen-science applications. Users must be first convinced to start participating. A second challenge is ensuring continued participation as the novelty wears off and users become disinterested. This becomes a greater issue as the requirements placed on the user increase. These motivations vary from person to person, may be intrinsic or extrinsic, and may relate to the interests and background of the individual. Mechanisms that incentivize users must consider these motivations, the value of user contributions, and handle abusive or exploitative use of the system. Adequately understanding user motivations and implementing appropriate incentivization strategies helps to get users participating in meaningful ways and to keep them interested in the long term.

Customizable privacy controls provided to users may promote participation by those more concerned individuals. These measures must strike a balance between allaying user concerns and providing high-quality data to the project. By sharing less information or by deliberately sharing less accurate data than possible we may improve the privacy of the user but reduce the benefit to the project. If we are unable to share the precise geographic location of a report but are instead able to provide only a coarse description we have reduced the value of this report. These controls may also interrelate with the incentivization scheme. Finding a balance between privacy options and report quality seeks to get as many users as possible participating whilst maintaining the best degree of precision possible at the same time.

The remainder of this work is organized as follows. Chapter 2 describes citizen science, its history, some notable applications and related works. Chapter 3 details two case study applications developed as part of this work. The role of context in mobile citizen science applications is discussed in Chapter 4. User motivation and incentivization is covered in Chapter 5. User privacy is considered in Chapter 6. Chapter 7 details the experimental evaluation and results. Concluding remarks and future work are contained in Chapter 8.

# Chapter 2

# Background & Related Work

Mobile applications for environmental and health monitoring are built off ideas relating to citizen science more generally and those associated with mobile applications as well. These exist in the intersection between the two ideas. Each come with their own set of challenges. While citizen science projects require attention to the ideas of context, user incentivization, and privacy, mobile applications place some additional constraints concerning network connectivity, battery life, and the limited user interface. Effective management of interaction context, user incentivization, and optional privacy controls benefit participatory sensing mobile applications for environmental and health monitoring by improving the quality and quantity of data shared.

## 2.1 Citizen science

Citizen science, or crowd-sourced science, brings the layperson into the scientific process as active participants. This typically takes the form of an individual assisting in the data collection, organization, categorization, or analysis processes of a scientific project [48]. The idea is that, in certain instances, even the uninitiated may contribute meaningfully to a scientific study. This involves, at the very least, a user volunteering some of their free time to participate. It may additionally involve the user spending some resources. The authors in [48] note that the burden involved may vary from project to project. Further, travel may be necessary or there could be urgent time-sensitive needs of the project that involve asking a user to go out of their way to participate. Both of these cases are increasing what is being asked of the user.

Citizen science has been applied to studying a wide variety of problems, many of which fall into the category of environmental monitoring. Some examples include: studying bird populations, tracking marine debris, monitoring bee population statistics, estimating the personal environmental impact, detection of potholes, gut microbiome diversity, galaxy classification, lake water quality, the discovery of prime numbers, the search for extraterrestrial intelligence, and many others.

The history of these projects is a long-standing one, although the term citizen science is more recent. An early example of this is the *Christmas Bird Count*, known as "the nation's longest-running citizen science bird project." This project was started on Christmas Day 1900. Today, tens of thousands of volunteers participate from December 14 to January 5 to report on bird populations. This data is shared with Audubon researchers, conservation biologists, and other parties [3].

The Great Internet Mersenne Prime Search (GIMPS) was founded in 1996 and allows users to contribute computational resources of their computers to the task of discovering new prime numbers. To date this project has led to the discovery of 15 additional Mersenne primes and in January 2016 found the largest known prime and the 49th Mersenne prime [2].

The Search for Extraterrestrial Intelligence (SETI) created a project that enabled users to have internet-connected computers contribute to the search by analyzing collected data. This project, known as SETI@home, was launched in May of 1999 [1].

A more recent project, the Marine Debris Tracker, seeks to monitor and track the presence of debris on the coastline. This project was started in 2010. Since inception the project has received reports by thousands of users and logged over three quarters of a million pieces of litter [4].

### 2.1.1 Sensing Applications

Many of these applications involve the reporting of some type of sensory information (visual, auditory, ambient temperature, and others). These applications are frequently found in the domain of environmental monitoring and are often enabled by the use of mobile applications. These measured sensory signals are shared with research projects that aggregate and analyze the reports. The reporting method can be broken generally into one of two possible categories: participatory or opportunistic. The choice depends largely on the particular application and desired trade-offs.

#### Participatory Sensing

Participatory sensing applications are those in which the user makes a conscious choice to meet some application requests [37]. This represents a typical approach taken by many applications, including the Christmas Bird Count, LAKEWATCH, Marine Debris Tracker, and Parkscan—a project for reporting problems with public park conditions. This interaction method requires that the user actively consider participation and fails if when they do not.

An example of this reporting strategy is the Marine Debris Tracker, which uses a mobile application to allow users to submit images of debris along coastlines [34]. The CyanoTracker application discussed in this work is also of the participatory sensing paradigm. These applications typically leverage human reasoning to identify and report data of interest.

#### Opportunistic Sensing

Opportunistic sensing applications are those in which reporting is done automatically when some state matches application requirements [37]. These applications wait until conditions are of interest to the project and report at that time. This may typically take place on a mobile device where periodic checks may be made of some sensor waiting for a desirable state change.

Challenges associated with opportunistic sensing include determining and adapting to the device sampling context, sensory coverage, and privacy [7].

An example of this type of application is the pothole detection application described in [23]. Instead of continuously uploading all acceleration data to the project, the accelerometers of the device may be constantly monitoring for the jerk experienced when a car hits a pothole and, upon detection, submit only that event. In this way we are able to reduce the communication overhead and participate in the project more effectively.

## 2.2 Mobile Platforms

Mobile applications for citizen science provide benefits that ease participation by users and may enhance the quality of user contributions. Smart devices may simplify user interaction by providing menus and icons from which a user can choose instead of requiring strictly text-based input [43]. Research shows that when data accuracy is very important and suitable software is employed, mobile devices provide a desirable platform for data entry [32]. The authors in [49] note that data entry into mobile devices is the next logical step after paper forms and personal computers. The ubiquitous nature of smartphones provides users an ever-ready tool for capturing and communicating data to associated projects. Modern devices provide a wide variety of built-in sensors. The authors in [42] note that mobile devices are becoming mobile sensory instruments by the inclusion of sensors for noise pollution, UV levels, water quality, and more. Participation is simplified through programming of convenience features and by automating the capture and input of data. Mobile applications are capable of implementing both participatory and opportunistic reporting mechanisms.

### 2.2.1 Challenges facing mobile platforms

The design of mobile applications for citizen science involves challenges including: hardware extensibility and compatibility, sensor calibration, capture and use of context, user incentivization, protection of privacy, limited battery life, a limited user interface, network connectivity, and bandwidth costs. In this work we examine three of these issues—context, incentivization, and privacy—through the development of two example applications concerning environmental and health monitoring. Effective management of interaction context, user incentivization, and optional privacy controls benefit participatory sensing mobile applications for environmental and health monitoring by improving the quality and quantity of data shared.

# Chapter 3

# Applications

The three challenges addressed in this work have been expressed through the development of two case-study applications: CyanoTracker and PillSafe. The CyanoTracker project seeks to address the environmental issue posed by blooms of harmful toxin-producing algae. PillSafe concerns medication safety as users report the desire to take an over-the-counter medication. This chapter details these two applications, providing background of the problem, implementation details, and presenting experiment results. In each application we consider the use of context, user motivation, and privacy protection.

## 3.1 CyanoTracker

The goals of the CyanoTracker project are to fuse the citizen science approach with traditional means of monitoring cyanobacterial growth. This includes a mobile reporting application as well as the use of social media platforms. Data from these two channels are combined with data obtained from a wireless sensor network (WSN) and from satellite imagery. This information is fed into an analytics system that seeks to provide an improved early detection and monitoring platform for cyanobacteria.

Figure 3.1 depicts an overview of the CyanoTracker system. It depicts the individual reporting components contributing to the project. Mobile devices in the top left communicate via cellular networks or via Wi-Fi when present. Other computers connect via the Internet. Satellite imagery, depicted in the top right, is fed into the system. Scientists provide the results of water sample testing (depicted at the

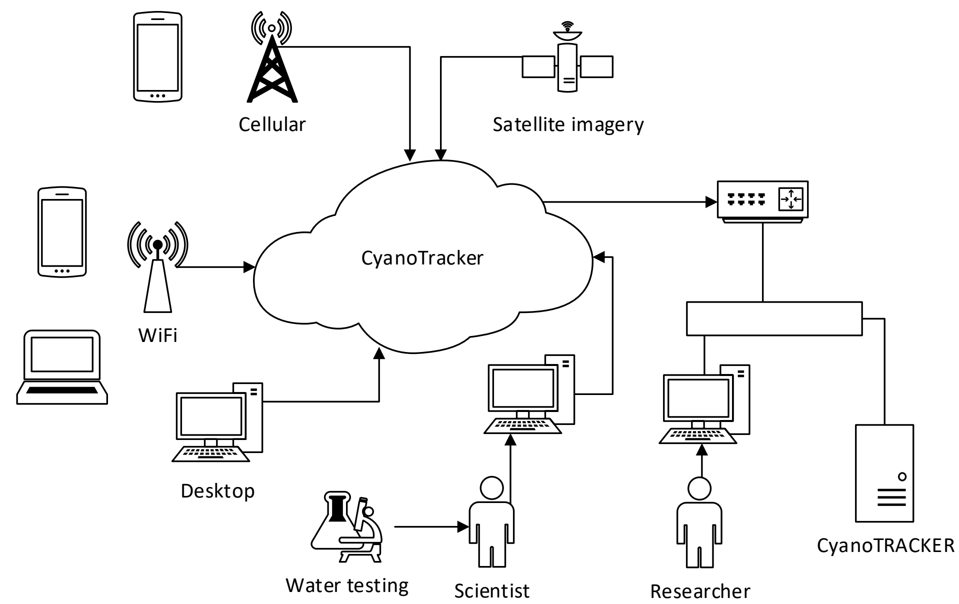


Figure 3.1: CyanoTracker system overview

bottom of the figure). The CyanoTracker service components are depicted on the right, consisting of servers, workstations, and participating researchers.

### 3.1.1 Cyanobacteria

Cyanobacteria, or "blue-green algae", are photosynthetic microorganisms capable of producing toxic substances. These toxins may affect the nervous system, liver, and skin [18, 12]. There are more than 60 identified toxins associated with cyanobacteria [36]. Their presence impacts water sources around the world, and may have a substantial impact on the ecosystem in both the short and long term [31]. Cyanotoxin exposure in freshwaters has affected humans and animals with illness, and even death, in over 50 countries and at least 36 states in the United States [29].

Table 3.1: Impact of Cyanobacteria, Indian River Lagoon, 30 year

| | |
|---|---|
| Tourism & recreation | $3 billion USD |
| Property values | $1.2 billion USD |
| Commercial fishing | $87 million USD |
| Total | ≈ $4.3 billion USD |

Data from the 'Save our Lagoon' project [46]

Several factors affect the size and growth of these populations, including sunlight exposure and nutrient levels in the water. While the absolute levels of nutrients matter the relative proportions are also important. Additional growth factors that may influence the potential for cyanobacterial dominance include CO2, pH, and salinity [41]. Explosive growths of these algae populations are known as harmful algal blooms (HABs).

In addition to the direct effects of cyanotoxin exposure, dissolved oxygen in the water is depleted as organic matter decays. This may secondarily lead to fish kills resulting from oxygen deprivation [12]. Birds may be poisoned by consuming affected vegetation and, in the case of carnivorous birds, through preying on affected animals [52, 40]. Livestock may become sick upon consumption of affected water. Fertilizer and manure run-off my nutritionally enrich teh water on farms [12].

Cyanotoxins have also been responsible for human poisonings and, in at least one documented event, fatalities [9, 10]. HABs impact nearly all water-based recreation activities, leading to concerns about the impact on tourism-dependent economies [12]. In Brevard County, Florida, for example, losses have been estimated at approximately $4.3 billion USD over a 30 year period [46], a breakdown of which is provided in table 3.1. Current environmental trends indicate that the problem is likely to worsen as climate change continues to progress [41].

### 3.1.2 Monitoring

Traditional means for monitoring cyanobacterial growth include: visual inspection, water abstraction and laboratory testing, satellite remote sensing, and uv spectroscopy. Each of these approaches is associated with certain strengths and weaknesses and may require varying degrees of expertise and access to equipment.

The traditional approach for monitoring cyanobacterial activity is for a site inspection to be performed to evaluate the water for discoloration, scums, and water clarity. The water may also be tested for parameters such as levels of nitrate, ammonia, and phosphorus. Water samples may be inspected to identify the microorganisms present. These techniques may involve the use of specialized equipment in the field or laboratory such as: Secchi discs, depth samplers, submersible probes, and photometers. [12].

A citizen science project known as the Florida LAKEWATCH program has volunteers collect samples of water for sending to labs for testing. These volunteers must undergo a short training process prior to participation which increases the requirements placed on the user. This program has, however, found that the values for total phosphorus, total nitrogen, chlorophyll content, and other crucial parameters were correlated strongly between samples taken by professionals and those taken by trained volunteers [8].

Remote sensing of cyanobacteria presence is possible through the use of satellite imagery. Phycocyanin, a chemical found in cyanobacteria, exhibits spectral reflection and absorbance patterns which may be used for detection. An absorption feature near 630 nm and a reflectance peak near 650 nm are used successfully by the MERIS satellite for the detection of phycocyanin [36]. One drawback to these approaches is the sampling period associated with the orbiting satellites. For example, MERIS has global coverage every 3 days [5]. Additionally this mechanism is not capable of use as an early detection system as the biomass required for detection by MERIS is above the threshold already considered as a bloom. Additionally, Kahru notes in [35] that detection of cyanobacterial blooms by satellite is possible with the formation of dense accumulations near the surface of the sea. Another satellite imaging sensor, known as MODIS, has an advantage of a daily revisit time, but the drawback of a coarser spatial resolution. The authors in [11] argue for "fusion" of data from MODIS and MERIS that outperforms traditional models for predicting microcystin distributions in Lake Erie.

These spectral properties may also be used for detection by deployed sensors on a body of water or by hand-held spectrometer. These may be deployed as part of a wireless sensor network that monitors several locations of interest on a body of water. A benefit to this approach is the specificity with which we may locate the devices and the frequency with which measurements may be taken. To be robust these approaches must handle significant variability in several parameters of water quality [44]. Models trained on samples collected from one site may have difficulty being applied elsewhere if they are not robust to these additional characteristics. In addition to fixed-location sensors, it has been suggested to make use of autonomous vessels which may sample from an increased coverage area [33, 20].

### 3.1.3 CyanoTracker

The CyanoTracker project seeks to augment the traditional methods of detecting and monitoring cyanobacteria with a citizen-science based reporting tool which allows users to submit photographs and reports of suspected HABs. These photographs are ideally location-enabled and timestamp-embedded to provide the most detailed information possible.

This application is implemented as a traditional client-server architecture, with a web service and corresponding mobile applications for Android and iOS. The app allows the user to capture photos or to submit previously captured photographs along with any necessary text or additional information. These reports may serve as an early detection mechanism as users visiting bodies of water are encouraged to report any suspect algal activity.

The goal of this application is not to replace traditional means for detection and monitoring, rather to augment them. The reports gathered by the citizen-science application may then be used to drive the application of other, more precise, and perhaps more expensive, measurement techniques.

The design of this application must consider a few peculiar features of our problem. This includes the potential for poor network access at reporting locations, report sparsity, and irregular and asynchronous report intervals. We address these challenges through application features such as queued report submission and incentivization mechanisms.

### Service implementation

The CyanoTracker service is implemented as a set of web servlets that accept reports from clients and maintain a database of submitted reports. This service then supports querying of this data by involved parties and may be used for analysis and mapping of cyanobacterial activity. Photographs are stored for potential future processing. A depiction of the service implementation layers is given in Figure 3.2.

### Client implementation

The CyanoTracker clients are implemented as mobile device applications available on both the Android and iOS platforms. These are deployed as native “apps” released in the respective marketplaces as opposed to a strictly web-based interface. The Android version currently supports devices of version 4.4+ and the iOS version 9.3+. The clients additionally make use of the Fabric/Twitter APIs and Google APIs.

| |
|---|
| Web Services |
| Servlet Container |
| Jave Entity Objects |
| Hibernate Persistence |
| SQL Database |
| Filesystem |

Figure 3.2: CyanoTracker Service Layers

The Android client additionally makes use of a networking library known as OkHTTP which is used to handle communication with the server. The Apache Commons IO library is used to provide various file input and output operations. The aFileChooser library provides some additional functionality. The Android application APK is currently under 5 MB. Care has been taken to ensure that as few permissions are required of the device as is possible to avoid needlessly alarming users considering installation.

The iOS client has been developed for phones and tablets running iOS version 9.3 and above. It has been written in the Swift programming language. A networking library known as Alamofire is used for handling communication with the server. The iOS version of the app is approximately 15 MB. The primary difference in application size here is the inclusion of separate image resources for various additional screen configurations.

In order to reduce incurred network costs, be robust to network disconnection, and to reduce energy expenditure, the app may be designed so that reports collected in the field may be reported later, asynchronously. This may be when the user has a better network connection, has a WiFi connection, plugs the device into a charger, or perhaps other triggering events. This client will maintain some user identification or authentication and transmit this along with reports. Users should be able to authenticate to a variety of popular third-party or social media platforms for ease-of-use while also providing a potential channel for communicating awareness of the project. For example, a user might be encouraged to tweet about their participation in the project after submitting a report to CyanoTracker.

An example of the user interface of the application's main screen during the reporting process is depicted in Figure 3.3. There is a short message about the project as well as buttons to use third-party authentication mechanisms provided by Twitter and Google. Users are also allowed to participate based solely on their provided email address.

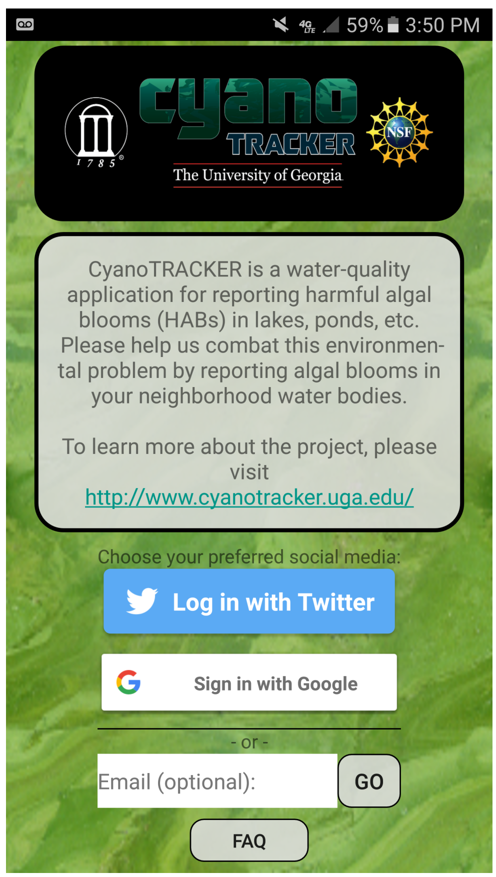


Figure 3.3: CyanoTracker application screenshot – main screen

Figure 3.4 depicts the user interface of the application's data entry screen during the report process. This example shows the interface with after taking an image which provides the contextual features of the location and time & date of capture. This information was automatically populated by the context handling procedures of the application. This screen also allows the user to provide any additional details they feel are relevant to the report. The user may finally submit the report or clear the form and start the report process over.


Figure 3.4: CyanoTracker screenshot – report screen with automatic context

An example of the reporting process when there is no contextual information available or permitted is given in Figure 3.5. In this configuration the user was required to enter each field individually. For the date entry there is a popup calendar which simplifies and improves the accuracy of the capture compared to manual text entry. The location precision is decreased here compared to the GPS captured values as the user was required to enter the location to the best of their recollection / understanding. This provides a much more coarse measure of the location.

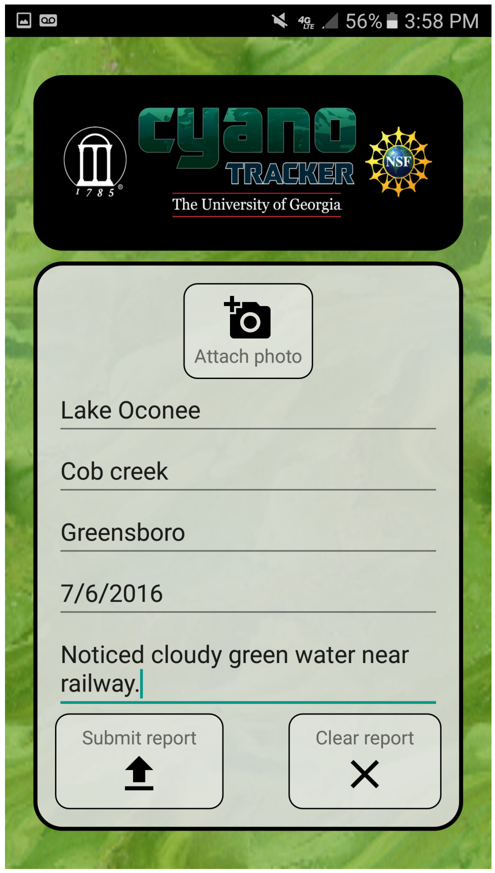


Figure 3.5: CyanoTracker screenshot – report screen without captured context

These screenshots depict a few possible examples of the user interface provided during the reporting process. Other parts of the application include a frequently asked questions (FAQ) section and a page thanking the user for their participation after a report has been filed. Later iterations of the application also allowed users to determine how many points they had accumulated through their participation in the project.

### Reports

The basic requirements for a report to the CyanoTracker project are minimal. An observation location and date are mandatory. Users may include a photograph, which is ideally location-enabled and containing metadata regarding the camera itself and the capture date and time. The basic outline of the report filing process is listed in Figure 3.6.

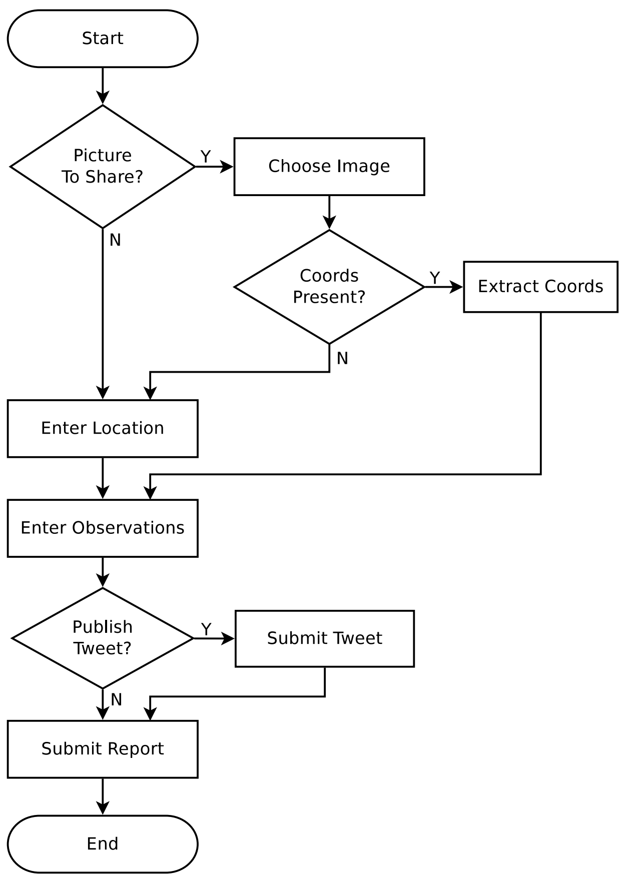


Figure 3.6: CyanoTracker Report Workflow

A simplified data model of a report is depicted in Figure 3.7. One point to note here is that a report may have several associated observations. This may include the initial report filed by a user as well as follow-up reports made by researchers or field technicians as additional detection and monitoring techniques are applied.

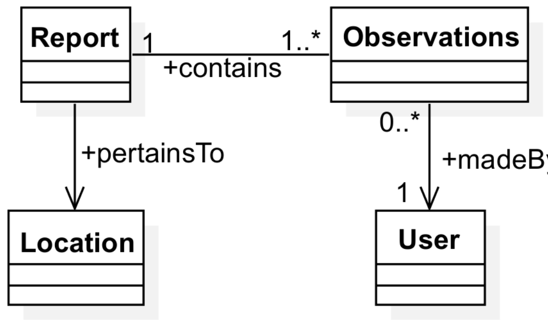


Figure 3.7: Simplified object model

## 3.2 PillSafe

### 3.2.1 Introduction

The Food and Drug Administration (FDA) defines a *medication error* as "any preventable event that may cause or lead to inappropriate medication use or patient harm while the medication is in control of the health care professional, patient, or consumer." [27] These errors are made by both patients and their caregivers and may occur in a variety of settings from the emergency department to the home. A 2007 figure estimates the costs of preventable medication errors (across all settings) at almost $4 billion USD [24]. The FDA notes in [26] the receipt of over 95,000 voluntary reports of medication errors between the years 2000 to 2009.

Over-the-counter (OTC) medications require special consideration due to their high accessibility and lack of required professional oversight during use. This leaves patients and their caregivers with the responsibility of ensuring safe medication use. In [47, 39], it has been shown that patients, particularly those with low health literacy or socio-economic status, may have difficulty understanding drug safety information. The authors in [50] argue that the language used in drug warning labels is at an inappropriately high reading level, potentially making them inaccessible to some members of the general public.

These matters are made worse by the fact that some products may contain more than one active drug ingredient, adding complexity to the safety determination process. Some drugs are present in a large number of differently named and labeled products. Acetaminophen, for example, can be found in over 600 prescription and OTC products [38] and may also be referred to by the additional names: APAP, Paracetamol, and dozens of brand names. Users must be aware of, and account for, all drugs administered from all products taken. The authors in [21] detail how users and caregivers may be unaware of, or seriously underestimating, the potential hazards of nonprescription medications. Some users appear to consider the availability as an OTC medication as an signal of inherent safety.

Three factors contribute to the potential growth of this problem: the large proportion of adults using OTC medications for themselves and their children [13], the introduction of new over-the-counter products [28, 13], and the ongoing desire to reduce healthcare costs. Without intervention, as the population grows, the impact of medication errors is likely to increase. This assumes the same percentage of users continue to use these products and do no change their medication safety habits. The introduction of new OTC products, whether a completely new product or one transitioning from prescription-only status, may lead to increased consumption of over-the-counter products. As the necessity to decrease healthcare costs continues to grow, addressing expenses related to medication errors becomes increasingly important, particularly as the problem grows.

### 3.2.2 Background

Efforts have been made on several fronts to reduce medication errors. Doctors and pharmacists have employed clinical decision support systems, which scrutinize medical treatments to ensure proper care, for years. Consumers may not have access to equally powerful tools to assist them in the safe use of OTC medicines. Such systems would consider pertinent aspects of a patient's health record as well as the published medication safety rules. There are static drug-drug interaction checkers available that do not consider the user health profile nor the passage of time between two drug administrations. Examples of these applications are found in [15, 22, 51, 19].

Meanwhile, efforts targeted to the consumer have included improved product warnings, labeling, and packaging. Additionally, there have been ongoing efforts to improve public awareness of OTC safety issues [26, 25]. These efforts have been implemented primarily on an as-needed basis to address acute situational needs [45]. Manufacturers themselves have participated in efforts to improve OTC safety in the interest of public safety. An example of this was the voluntary removal of certain cough and cold medications (CCMs) marketed for infants from the market. Additionally the packaging on all over-the-counter CCMs was updated to contraindicate the use in children under four years of age [30].

### Rule Engines

Rule engines provide an abstraction for decision making in a system. They provide a mechanism to separate certain behavior rules out of a system in a way that simplifies the addition, removal, and changing of these behaviors without the need to modify the code of the system itself and, therefore, do not require recompilation to make these changes.

Rule engines have been applied in many different areas in the past. They are frequently employed in business settings to provide a higher-level abstraction for decision-making. This allows those without programming experience to affect the decision making process by authoring rules in a more digestible language. In the

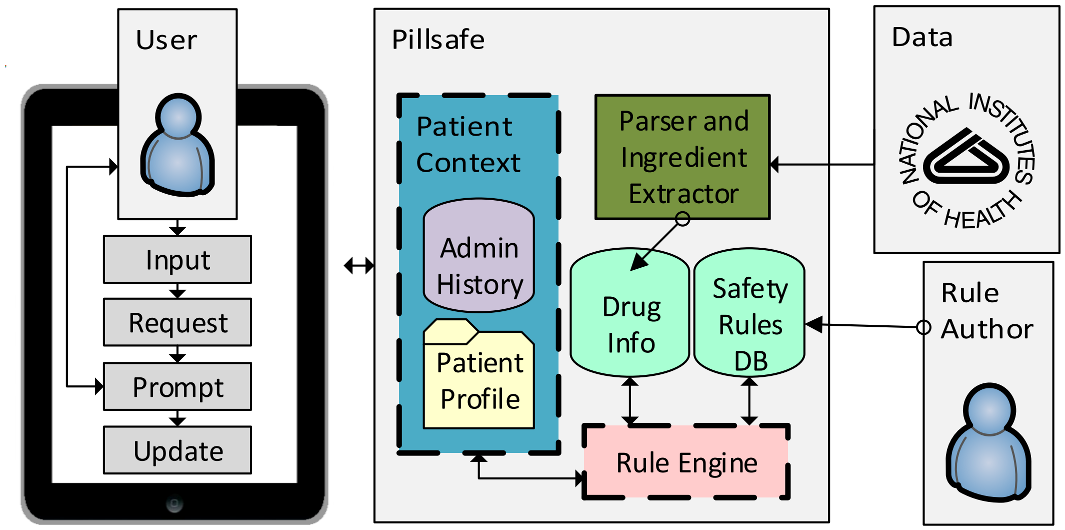


Figure 3.8: PillSafe application architecture

PillSafe project there is a similar requirement. Ideally those entities in the pharmaceutical industry, medical, and regulatory agencies could author rules in a language tailored suitably to this purpose. The input from these domain experts is then fed into the rule engine along with information about a user.

### 3.2.3 Architecture

The PillSafe system architecture is depicted in Figure 3.8. Drug data originates from authoritative sources such as the National Institutes of Health. Currently, rules are input to the system by special users designated as rule authors. The PillSafe service maintains a database of drugs and their safety rules. It also may optionally contain a rule engine responsible for executing safety queries and the relevant patient information. In other configurations, the patient data and rule engine may be located on the mobile device. This is done in an effort to protect patient privacy.

The tradeoffs associated with the placement of the patient data and rule engine are outlined in Table 3.2. Considering the fact that there are two objects to place and each has two possible locations we have a total of four possibilities to discuss. The lowest bandwidth requirements are associated with configurations that have the patient data and rule engine in the same location. When patient data is stored on the device while the rule engine exists on the service side, it is necessary to transmit all pertinent aspects of the patient profile. The same is true when the patient data is maintained by the service while the rule engine is operated locally.

Privacy is most protected when patient data and rule execution occurs on the device. This reduces the exposure of patient medical data and the information of the medications being requested. We strike a middle ground when patient data resides on the device but the rule execution occurs at the service level. It may not be necessary for the service to have the patient data in its entirety. Instead, a rule might ask whether the patient suffers from hypertension; a query that may be answered by the device without the need to reveal a list of all conditions affecting the patient. With the patient data stored centrally and the rule engine operating on the device another tradeoff is struck in terms of privacy. In this configuration, while the service has access to the patient profile, it does not receive information about every advice the patient seeks. It will, however, be updated when a patient confirms they are indeed taking a medication.

Although it reduces the notion of user privacy, maintaining the user profile at the service level provides some significant benefits. The foremost is an improvement in data consistency when users interact with the PillSafe service through multiple devices. Multiple independent histories across devices may lead to inappropriate safety advice being generated by the system. Maintaining either object on the service side has the unfortunate effect of breaking functionality in the event of network connectivity loss.

Table 3.2: PillSafe Design Choices

| | Local State | Remote State |
|---|---|---|
| Local Decision Making | Low bandwidth<br>Most private<br>Potential loss of state | Higher bandwidth<br>Middle-ground for privacy<br>Potential connectivity loss |
| Remote Decision Making | Higher bandwidth<br>Middle-ground for privacy<br>Potential loss of state | Low bandwidth<br>Least private<br>Potential connectivity loss |

```
(defrule 0363-0007-03_condition_pregnant
      "If pregnant, ask a health professional before use."
      (User (OBJECT ?us))
      (test (?us hasCondition "pregnant"))
      =>
      (add (new Advice (SafetyStatus.CAUTION) "If
         pregnant, ask a health professional before
         use.")))
```

Figure 3.9: Example JESS rule pregnancy rule

### 3.2.4 Implementation

A prototype realization of the PillSafe system has been implemented in order to conduct experiments. In this implementation the rule engine is maintained exclusively on the server side. Experiments have been performed to test both the device-maintained state as well as centrally stored. The first experiment was to measure the response times and to determine whether they were reasonable for user requests. These tests were conducted with varying connection qualities and with varying complexities of user health profiles. Secondly the bandwidth requirements of such a system as it relates to the length of a user history.

### Details

The server portion of the PillSafe system has been implemented using Java servlets and was deployed using Apache Tomcat version 7. Services are provided through RESTful web services. The service was hosted on an 6-core 3.2 GHz AMD machine with 8 GB of RAM. Requests were made from ethernet, Wi-Fi, and cellular network connections in order to test the feasibility of use of the application on these various connection types.

The client is currently implemented as an Android application. The client application connects via Wi-Fi to the Internet or through the use of cellular networks (if available). Messages are exchanged between the client and server using JavaScript Object Notation (JSON) objects.

The decision-making component of the PillSafe system is the Java Expert System Shell (JESS) rule engine. This component interacts with Java objects to describe conditional behavior on these object properties and associated actions to perform. Recent versions are also capable of running on Android devices. Later we give brief examples of PillSafe rules implemented using the JESS rule language.

### Data

Data for this project was gathered from the DailyMed Structured Product Labeling (SPL) database, which contained 74,112 listings at the time of original implementation [16]. These listings contain all required product labeling information for OTC products in specially formatted eXtensible Markup Language (XML) documents.

Considering the effort required to manually author rules for the system, a set of 24 commonly used drugs was chosen. The complete list of these chosen drugs is found in appendix A. For each of these, a representative product was chosen. This was required as some drugs are found in a very large number of products. For example, an initial search for acetaminophen returned over 3,900 results. The chosen product was also allowed to contain additional active drug ingredients. In order to further narrow the search focus only solid-dosage products from well-known brands were chosen for consideration.

Relevant sections of each document were then extracted and used for further processing. This included: the NDC code, product packager, active ingredients list, administration rules, and safety warnings. The set of allergies, health conditions, and prescriptions learned through processing of these files constitute the basis of knowledge of the system.

Each product was then manually codified, as much as possible, in PillSafe using rules expressed in the JESS rule language. Some exceptions were made for rules that were particularly difficult to represent or implement in the system.

Several user profiles were created for use in testing the system. The three primary test users were created to reflect patient profiles of various sizes and complexities. The first user had no recent drug administrations and took only one prescription drug. The second user had 8 recent administrations and one disease on record. The third user had 15 recent drug administrations, two health conditions, and one known allergy. The few remaining users had much longer histories created for testing purposes.

### Rules

Figure 3.9 depicts an example rule which states that pregnant users should consult health professionals before taking Omeprazole (NDC #0363-0007-03), a medication for treating heartburn, stomach ulcers, and gastroesophageal reflux disease.

One key design goal of the PillSafe system was to provide a domain specific language (DSL) for specifying safety rules. The object here was to enable experts from domains other than computer science, such as those in the pharmaceutical industry, to author rules using a simple and easy-to-use language. These rules should also

```
$caution condition: pregnant, breast-feeding
If @, ask a health professional before use.
```

Figure 3.10: Higher-level representation used to generate JESS pregnancy rule

```
(defrule 0363-0640-08_age_range_4_12
      "Children 4 to under 12 years of age: ask a doctor
         for directions."
      (User (OBJECT ?us))
      (test (and (>= (?us getUserAge) 4) (< (?us
         getUserAge) 12)))
      =>
      (add (new Advice (SafetyStatus.CAUTION) "Children 4
         to under 12 years of age: ask a doctor for
         directions.")))
```

Figure 3.11: Example JESS rule considering a user's age

be ready for use in a rule-engine, making the rules themselves directly executable. Figure 3.10 shows the same Omeprazole pregnancy rule written in the PillSafe DSL.

Some rules are more complicated than the presence or absence of health conditions or other drugs. There may additionally be restrictions based upon an individual's reason for taking a medication. For example there is a rule from an Ibuprofen-containing product (NDC #10202-647-62) that states "Ask a doctor or pharmacist before use if you are taking aspirin for heart attack or stroke, because ibuprofen may decrease this benefit of aspirin." In this case we must consider the reason a drug is being taken.

```
$caution age_range: 4 12
Children 4 to under 12 years of age: ask a doctor for
   directions.
```

Figure 3.12: Higher-level representation used to generate JESS age rule

```
(defrule Acetaminophen-overdosage
      "Acetaminophen overdose"
      (User (OBJECT ?us))
      (test (?us exceedsDrugDoseInWindowHours
         "acetaminophen" 4000 24))
      =>
      (add (new Advice (SafetyStatus.WARN) "Use of this
         product would result in acetaminophen overdose")))
```

Figure 3.13: JESS acetaminophen dosage cooldown

An important and frequent consideration for taking medications is that one must not exceed a certain dosage in some time frame (often defined in a 24-hour time period). This is a slightly more complicated predicate that requires examining the recent administration history recorded. An example of this type of rule using the JESS language is depicted in Figure 3.13.

# Chapter 4

# Context

The effective capture and use of interaction context benefits mobile applications for participatory sensing citizen-science applications by reducing the burden placed on the user and improving the quantity and quality of data reported. At the most basic level a report consists of some sensory data to be shared. We may combine this core sensory information with input from various other sources as well as metadata about the captured data. An example of this is depicted in Figure 4.1.

An important tradeoff exists between the context capture and the resource consumption. For example, acquiring data from the various sensors on the mobile device may incur some additional costs. However the utility of the data submitted as a result is improved. In general contextual information should be automatically collected when available.

A second tradeoff exists between the context capture and the user effort required for participation in a project. As more data is automatically collected and reported there is less for the user to do. This tradeoff improves reports by providing additional data input to the report process. Reducing the user effort helps encourage participation.

This data may come from a variety of sources but the idea is that it should not require additional user interaction to include. It is up to the system to manage and include relevant details. This data may come from additional sensors built into or connected to the device or from external data sources which may be queried at report time or possibly after-the-fact and may be collected by the device or on the

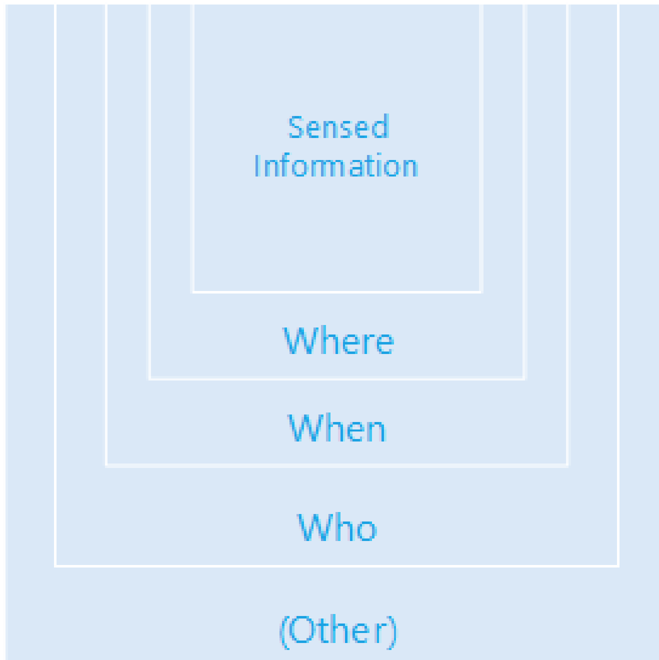


Figure 4.1: Contextual information surrounding report

service side. The benefit of performing the additional lookups on the service side is the reduction in resource demands placed on the mobile device. A depiction of this setup is given in Figure 4.2.

One issue facing development of these applications is the potential for heterogeneity across devices. Devices may include or exclude certain sensors expected by the application. Additionally the degree of precision may vary across devices. The software must be capable of handling these in an application-specific way that seeks to promote the project goals.

The next sections detail the notion of context as it applies to the two example applications. Each section lists the various aspects of interaction context we are interested in and how they are obtained. Fallback mechanisms are discussed to robustly handle devices with a variety of sensors available. By effectively managing the interaction context with citizen-science projects we seek to reduce the input required by the user while improving the precision of the reported data.

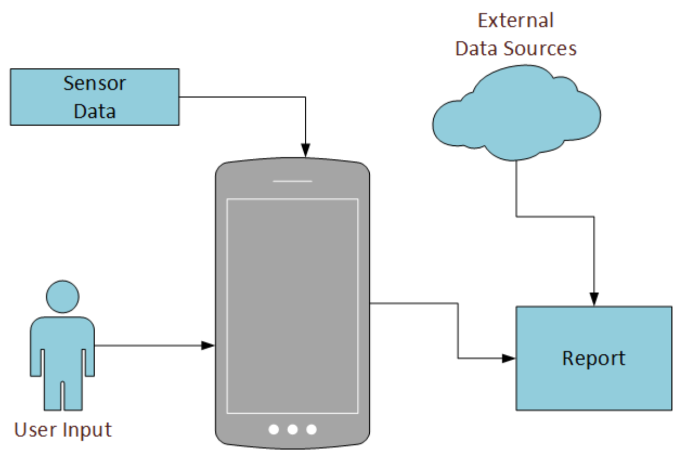


Figure 4.2: Report context sources

## 4.1 CyanoTracker

An interaction with the CyanoTracker service is, at its core, a signal representing a suspected sighting of cyanobacteria. It is around this message that we attach as much pertinent data as possible in order to enhance the value of the user participation. The interaction context in the CyanoTracker project that provides useful information includes the device GPS coordinates, time of capture, camera properties, and user authentication. These contextual features are automatically supplied by the application. This helps to alleviate the burden on the user and, in turn, helps to improve the overall quality of reports.

Some additional contextual information may be obtained after report submission and are thus not required by the device. For example, information regarding the weather at a reporting location may be obtained by querying historical data provided by the National Centers for Environmental Information (NCEI), a service by the National Oceanic and Atmospheric Administration (NOAA), under the U.S. Department of Commerce. Due to this easy availability the mobile device does not collect this information. Figure 4.3 depicts the basic signal in the CyanoTracker application as well as the additional context included.

**Cyanobacteria report**

**Signal:** I suspect cyanobacteria activity.

**Context:**
Report timestamp
Device location
Camera properties
User authentication
Historical weather data

Figure 4.3: CyanoTracker example cyanobacteria report format

There are challenges involved with integrating this context. For instance, some users may deny the ability of the phone to query the GPS coordinates. In this case we simply take a short text description from the user and accept the reduced precision.

If the user is unwilling to capture and submit a photo through the application we are also unable to automatically obtain an observation date and time. In this case the user can enter the time value manually or specify that the report they are filing is a current observation in which case we may simply grab the time from the device during the report process.

The CyanoTracker project supports authentication to third-party organizations such as Google. However, in order to promote user participation the application still allows users to participate in a (semi) anonymous fashion, not requiring the authentication to any external system. In this case the metadata associated is not related to the third-party authorization but rather some details of the mobile device itself.

## 4.2 PillSafe

The interaction with the PillSafe project is, at its core, a signal stating "I'd like to take medication X right now." The context of an interaction with the PillSafe project involves several features. These relate to the user's health conditions, allergies, medication history, and other elements of the electronic health record. These items are maintained by the application in such a way as to automatically provide any details possible with the goal to reduce the required input and burden on the user. Medication administration history is automatically recorded as the user interacts with and notifies the system. Considering that proper accounting of all medications has been previously identified as a particular challenge, this is a primary benefit of the use of this system. The capture and use of context in PillSafe help to ease the burden placed on the user as well as to improve the accuracy of the data that, in turn, improves the advice returned by the system.

A query to the system is automatically timestamp-embedded by the device so that the user does not have to enter this. The results of safety queries are also stored and become part of the context of the user interactions. That is, if the user was warned not to take a medication as it would be unsafe and they take it anyway both of these points are noted. The history of administrations becomes part of the context automatically and is built on the fly during interactions with the system. Figure 4.4 depicts the basic message format for this interaction as well as the various contextual elements involved.

A challenge facing this system is encountered if users interact with the system from various devices. In this case it is required that we may ensure consistency across these devices. An example of this is illustrated by the following scenario: a user

**Over-the-counter medication request**

**Signal:** I'd like to take medication X right now.

**Context:**
Report timestamp
Health conditions
Allergies
Medication history
Warning history

Figure 4.4: PillSafe example medication query report format

takes medications while at work using PillSafe on their cellphone and later accesses PillSafe from a tablet at home. If the user took ibuprofen a few times during the day but this information is only registered on the cellphone, safety queries made later at night from the tablet may be unable to successfully complete safety determinations as it only has partial information.

The system must be designed in such a way as to handle this use case. Another challenge facing this system is the potential to travel through various time zones. We must be careful to ensure that this does not affect safety queries by allowing medication administration too soon. We may accomplish this by moving everything to one magazine.

The user electronic health record may originate from a healthcare professional and describes the patient's health conditions in detail. This record may also include all the prescription and non-prescription medications being taken by the user. This is more effective and complete than expecting the user to make a complete accounting for their health status.

An example of the information relevant to user queries, including contextual information, is demonstrated in Figure 4.5. By automatically constructing and including as much information as possible we simplify the interaction with PillSafe whilst also increasing the quality of the data being sent by the user. This, in turn, helps to ensure the quality of query responses.

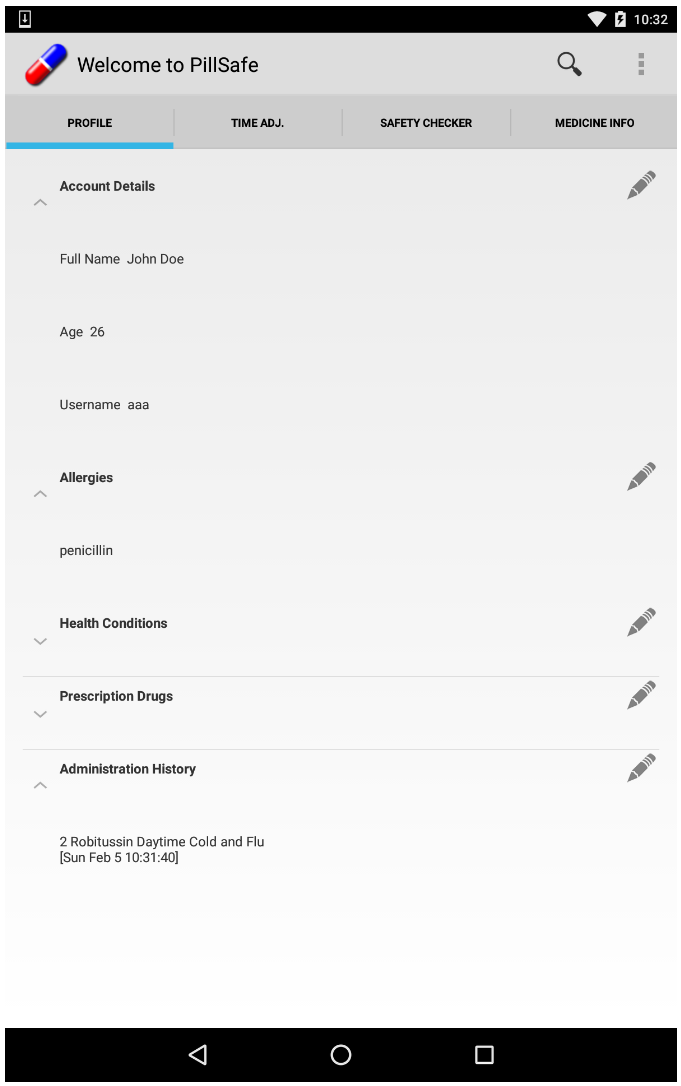


Figure 4.5: Pillsafe patient profile

# Chapter 5

## Incentivization

Incentivization is a particular challenge facing citizen science applications. We must consider a user's motivations for participation. This applies to both the short and long term; we need to convince users to initially join in on a project but further we hope to find ways to engage users for continued cooperation. Motivation can be both intrinsic and extrinsic and may vary from individual to individual.

Intrinsically motivated users are those for whom we may not have to do much work to elicit participation. These users might be concerned about the ecosystem, public health, improvement of public resources, or simply desire to contribute to the scientific process. In the case of PillSafe, a user might be motivated by concern for their safety. Participation in CyanoTracker may require a bit more motivation, as users may not feel the problem affects them directly. The greater the geographical distance from a user and the greater the time delay before an impact is felt both may decrease the perceived urgency felt by individuals. The more relevant a project's impact can be made to users the greater the effect of this personal motivation. Extrinsically motivated users may require some external rewards or incentives to convince them that their participation is worthwhile. Winning over these users is the goal of gamification mechanisms in citizen science applications.

Incentivization schemes, or gamification mechanisms, seek to improve this situation by providing some rewards for participation. These must consider: the components of a valuable report and their relative importance, the motivations of participants, and the resource expenses incurred by users.

To design an incentivization scheme, the individual elements that affect the quality of a report in that application must be considered. In general as the volume of data reported increases or as the precision and accuracy of data reported increases the value to the project is improved. In environmental monitoring systems reports from areas not reported on recently may be worth more than those for which reports are often received. Users who routinely submit useful, high-quality reports may have future reports valued more highly. The relative importance of these features as they relate to the project goals must be considered.

The motivations of participating users must be considered on an project-by-project basis. For example, more intrinsically motivated users may be likely to submit higher quality reports and additionally may have a lower expectation of incentivization or rewards for their participation. Those more extrinsically motivated, on the other hand, may require adequate mechanisms to reward their participation and, more importantly, their submission of high-quality data.

Finally, some understanding of the sacrifices expected of the users should be arrived at. This will be balanced against the project goals and should be considered as it relates to the incentivization scheme. Any participation that requires significant expenditure might be promoted by increased rewards when compared to simpler cases. As one of the challenges facing the design and development of mobile applications for environmental and health monitoring, management of user incentiviztion benefits the quality and quantity of data shared.

## 5.1 CyanoTracker Study

In an effort to determine how users are motivated to participate in citizen science projects and how to best design gamification mechanisms to promote user engagement we have conducted an anonymous survey using the Qualtrics software. There were a total of twelve questions on this survey, each of which was optional. Some of the questions were demographic in nature, while others related to the motivations and incentivization to participate in citizen science applications.

These questions included: asking how often respondents have participated in citizen science projects, asking what motivating factors might have them participate in CyanoTracker, what types of incentives might motivate them, and etc.

We received 116 results from this survey from students, professors, staff, industry professionals, and farmers. A comparative study of each question was completed for each question by age group. Respondents were divided into six age groups. These were: 18-24, 25-34, 35-44, 45-54, 55-64, 65+. The respondents were also considered along gender lines. The results of this survey are discussed in the next section.

## 5.2 Survey results

The demographics of the respondents were as follows: approximately 40 percent of respondents were in the 18 to 24 year age range. Of these, about 73 percent of them were male while the remaining 27 percent were female. In this group about 95 percent of users were students, with about 5 percent as private industry employees. About another 23 percent of the respondents fall into the next age group, 25 to 35 years of age. In this group we find that about 46 percent of them were male whilst the remaining 54 percent were female. In the remaining age categories we find about 16 percent of respondents in the 35 to 44 year range, 9 percent in the 45 to 54 year range, 6 percent in the 55 to 64 year range, and about 6 percent in the 65 years and above [17].

Based on these demographic results it can be seen that we primarily reached younger adults. Many of these respondents were students. This may have had an effect on the likelihood and participation habits of engaging in citizen science projects reported by users.

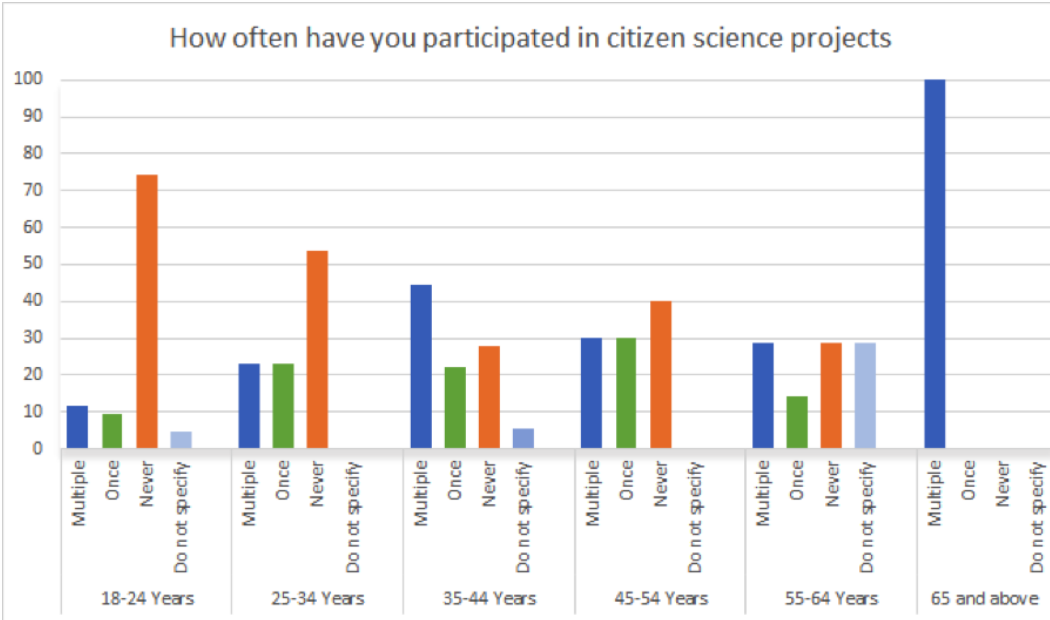


Figure 5.1: CyanoTracker survey results—past participation. Figure credit [17]

### 5.2.1 Past participation

Respondents were asked "how often have you participated in citizen science projects?" The results of this question are shown in Figure 5.1. The results indicate that as we look to older age groups we find users were more likely to have participated in one or more citizen science projects. The likelihood of a user having never participated in a citizen science project diminishes as we look to older age groups.

### 5.2.2 Likelihood of participation

The question was asked: "how likely are you to participate in CyanoTracker?" The responses available were: "definitely", "likely", "not likely", and "not sure." The results of this question are found in Figure 5.2. We find that as we look into the

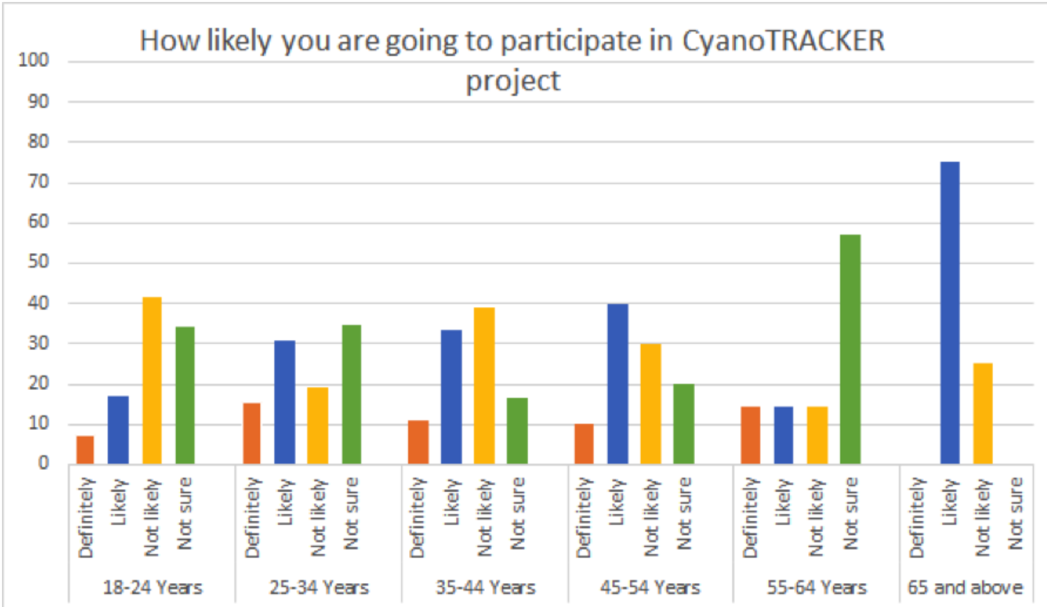


Figure 5.2: CyanoTracker survey results—likelihood of participation. Figure credit [17]

older age groups there are more and more users who will either "definitely" or "likely" participate in the project.

### 5.2.3 Motivations

Users were asked: "what motivates you to participate in CyanoTracker?" The possible responses include general concern about the environment, concerns about pet health and safety, concern for neighborhood ponds and lakes, some token gift or reward, and other. The results of this survey question are found in Figure 5.3. We find that, overwhelmingly, users were most motivated by a general concern for the environment. Secondarily users appear to be worried about local lakes or ponds. These bodies of water are more relevant to users due to their proximity.

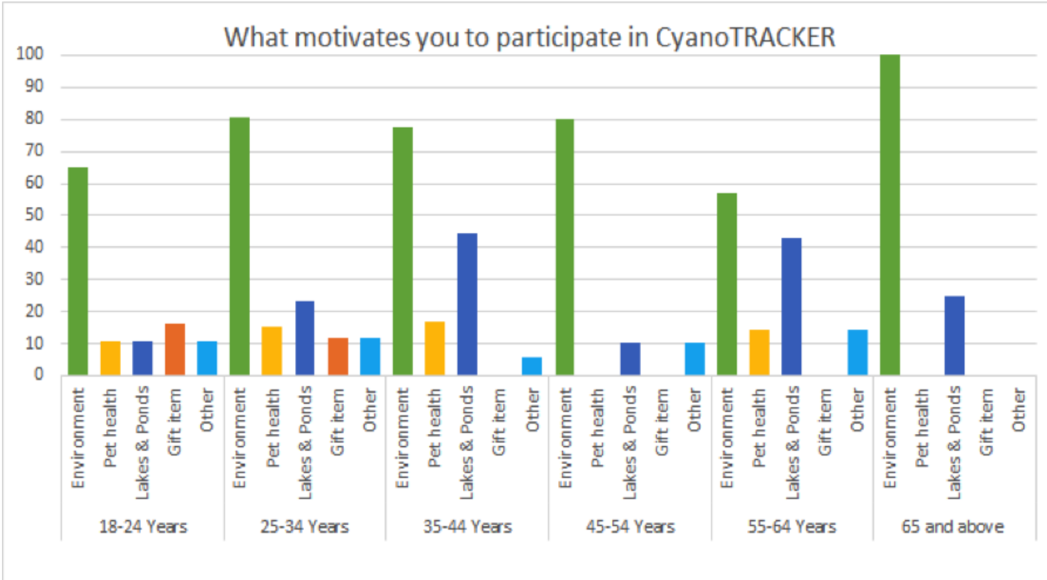


Figure 5.3: CyanoTracker survey results—participation motivation. Figure credit [17]

### 5.2.4 Effect of incentivization

Respondents were asked how their "level of participation will be affected if incentives are provided." The options provided include: "more", "the same", "less", and "not sure". The results of this survey question are seen in Figure 5.4. We find that users in the 18 to 24 year age group are much more likely to participate provided there are incentives provided. Older respondents tended to care less about the incentives. Many users over 34 appear to indicate they will participate the same whether there were additional incentives or not.

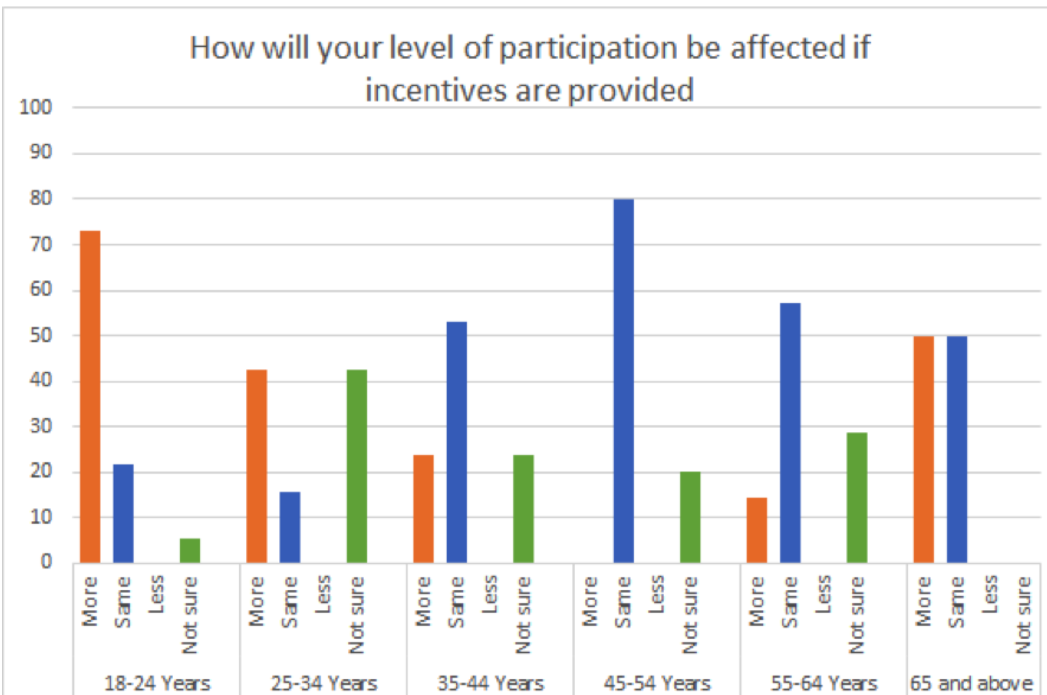


Figure 5.4: CyanoTracker survey results—effects of incentivization. Figure credit [17]

### 5.2.5 Incentive desirability

Users were also asked: "what types of incentives attract you the most?" The options provided included: a certificate of recognition, some type of credit counted towards coursework, recognition via social media, and other incentives. The other incentives listed by individuals included money, the reputation of the study, and the positive impact on the environment. The overall results of this question are seen in Figure 5.5. We find that users in the 18 to 24 year old range may be significantly more interested in earning some credit for classes for than other potential rewards. Courses concerning the environment or ecology might offer extra credit for participation in relevant citizen science projects. These users appeared to be less motivated by recognition by certificates or on social media [17].

### 5.2.6 PillSafe incentivization mechanism

In PillSafe, incentivization was implemented with lessons learned from the motivation survey. The basic principles established are that there should be some baseline points rewarded for participation. Features such as the first user report, location "cooldown" upon repeated reporting, and a recommendation system should then affect this reward. Additionally users have an associated "level" that they can improve over time. These components are described more fully in [17].

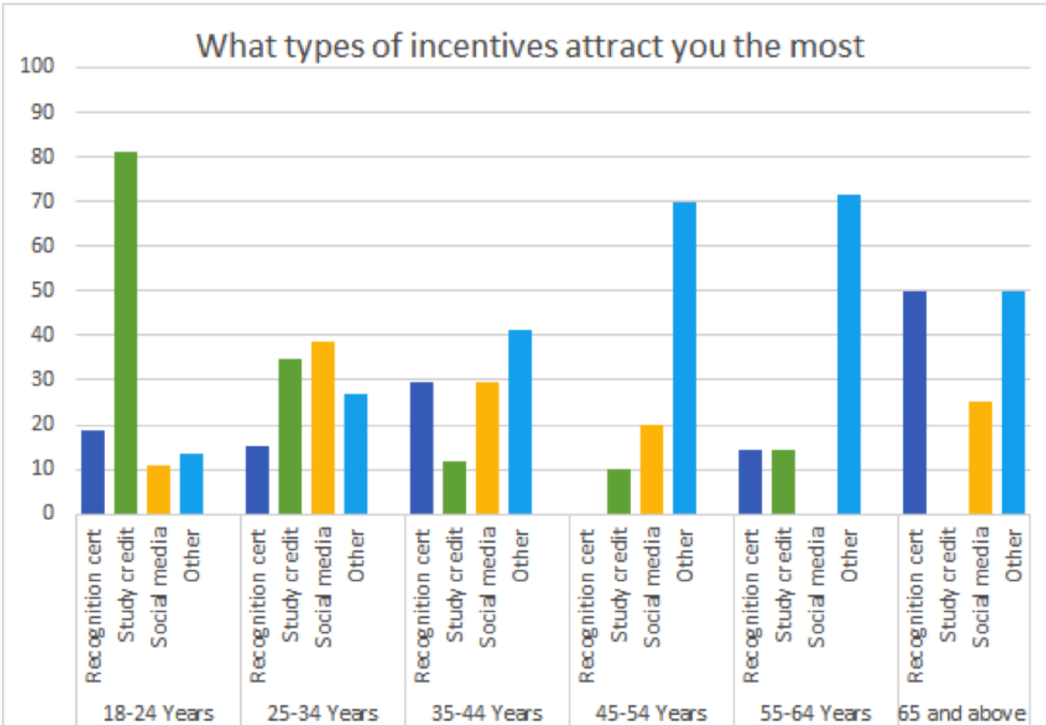


Figure 5.5: CyanoTracker survey results—desired incentives. Figure credit [17]

Table 5.1: PillSafe incentivization components

| Component | Points |
|---|---|
| Baseline reward | 5 points |
| First report | 60 points |
| 30th report | 50 points |
| Same lake from same user within 1 week | restricted |
| User level upgrade | 100 points |

# Chapter 6

# Privacy

Privacy is a primary issue facing citizen science applications. A tradeoff must be established between the utility of data provided to the project and the protection of user privacy in the process. This is essential to promote participation of as many individuals as possible. By providing privacy options that satisfactorily protect users' privacy, we reduce the number of potential users we turn away.

## 6.1 CyanoTracker

Privacy in CyanoTracker is an issue as user's geographical location is involved. Particularly privacy-conscious users may be uncomfortable participating unless there are privacy protection mechanisms in place. These properties ought to be configurable to address the varying privacy expectations users have.

A first improvement involves the ability to define a privacy area in which the user's location will not be reported. This area might be defined to include the user's residence or workplace. Any reports from within this area could be stripped of geolocation data.

When reports are filed, it is important to maintain geolocation features of the report and any submitted images. One way to improve privacy in this situation while maintaining data important to the system is to simply delay the report submission. This benefits a traveling user by never revealing where the user *is*, but instead may reveal where the user *was*. This applies except in those circumstances where the report comes from a predefined privacy area.

### 6.1.1 Point-of-Interest Notification

In the CyanoTracker project, there are specific locations of particular interest. These may be locations of cyanobacterial activity in the past, those which have not been observed recently, or may be suggested by predictive modeling. Instead of passively waiting for a user to happen upon the location, a desired feature of the CyanoTracker application is to provide users hints or notifications of nearby points-of-interest. The addition of this feature benefits both the user and the CyanoTracker group. The incentivization rewards are increased due to their increased value. The project benefits through receipt of this valuable data, while the users benefit by earning additional rewards.

A basic query to the CyanoTracker system of this type might ask: "what are the POIs within 25 miles of my current location?" Further, while a user is traveling they may be alerted to the presence of nearby valuable points of interest. The distance a user may be willing to travel could be customized to suit.

### 6.1.2 Privacy-preserving Point-of-Interest Notification

One potential issue here is that some users may be hesitant to share their precise location with the service. Servicing only nearby POI requests makes this a particular issue, as it requires some knowledge of the user's location. In order to meet these simultaneous demands, we first obfuscate the user's location by randomly offsetting their location, requesting POIs in a larger area surrounding this new location, and then allowing the device to do the final filtering based on the actual location and travel radius.

Additionally the service may not want to share all available points of interest with all users and instead only give information on those that are known to be nearby. This bi-directional restriction of information exposure benefits both parties.

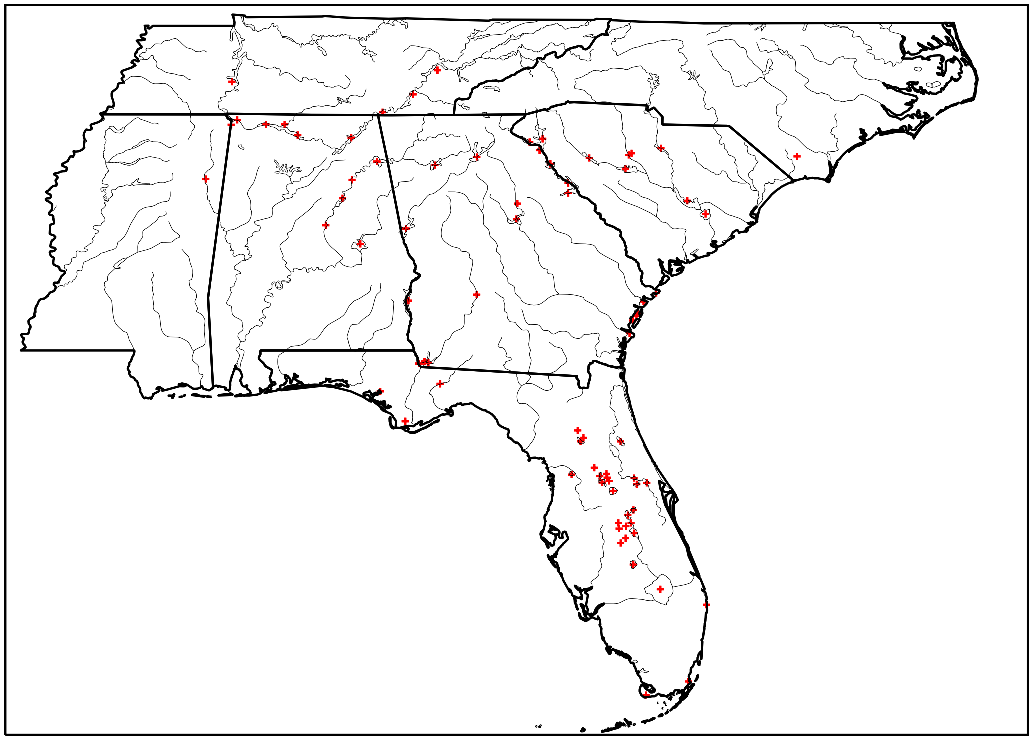

Figure 6.1: CyanoTracker Test Points of Interest

The code to determine points within a given radius from a selected point is depicted in Figure 6.2. This filtering can be done on the device to narrow the selection down from those in the privacy radius to those in the travel radius. The entire program listing is found in appendix.

A user's location is obfuscated by the pseudocode outlined in 6.3, where random() returns uniformly-distributed real values in $[0, 1)$ and a Point has both a lat and lon member. A random angle is chosen in $[0, 360)$ along with a distance in $[0,$ `privacyRadius`$)$. Figure 6.4 depicts these components. The smallest circle is the user's travel radius; this is the distance they may be willing to travel to fulfill a POI request. The mark at the center of this circle is the current location of the device.

```
def getNearby(allpoints, deviceLocation, radius):
    returnPoints = []
    for point in allpoints:
        dist = vincenty(deviceLocation, point).miles

        if dist <= radius:
            returnPoints.append(point)

    return returnPoints
```

Figure 6.2: CyanoTracker Nearby Location Filtering Code

The larger diameter circle that is still centered on the user's current location is the privacy area. This is the area from which a location will be chosen at random by the device. This location is then used as the center of the search area and is communicated with the service to determine nearby points of interest. The device itself then performs the final proximity measurements to determine which available points are, in fact, within the selected travel threshold. The pseudocode for this process is listed in Figure 6.5, where "device" provides information about the mobile device, "service" provides access to the CyanoTracker web service which answers POI queries.

```
def getObfuscatedLocation(theLocation, privacyRadius):
    angle = random() * 360.0
    distance = random() * (privacyRadius/degreeDistance)

    return (float(theLocation[latIndex])+
                math.sin(math.radians(angle))*distance,
            float(theLocation[lngIndex])+
                math.cos(math.radians(angle))*distance)
```

Figure 6.3: Location Obfuscation Pseudocode

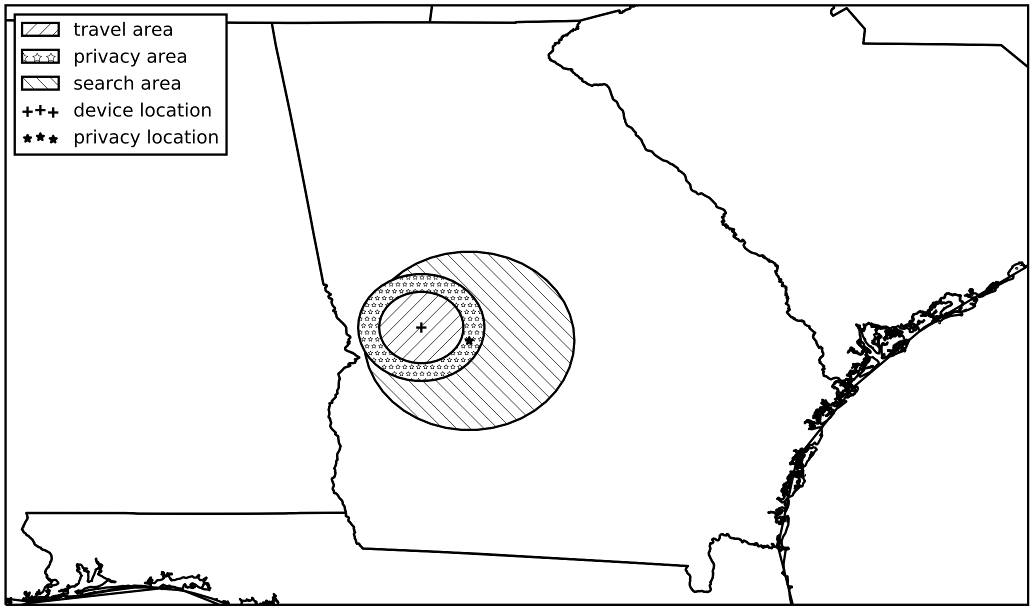


Figure 6.4: CyanoTracker Location Obfuscation Visualization

```
def privatePoints(allpoints, deviceLocation, travelRadius,
   privacyRadius):
    returnValues = []
    obfLoc = getObfuscatedLocation(deviceLocation,
       privacyRadius)
    privList = getNearby(allpoints, obfLoc,
       privacyRadius+travelRadius)
    return privList
```

Figure 6.5: Privacy-preserving POI Request

### 6.1.3 Experiment

In order to test this setup an experiment has been conducted which considers the overhead incurred by this privacy-preserving point-of-interest querying. This overhead is the number of POIs returned by the service which are within the user's privacy radius but not contained within the user's travel radius.

The experiment has been conducted on two synthetic data sets. In each, 300 points were chosen as points-of-interest from the southeastern United States. These points were limited to the following states: Tennessee, North Carolina, South Carolina, Mississippi, Alabama, Georgia, and Florida. In the first, all 300 locations were randomly distributed across this entire area. The points generated in the first dataset are depicted in Figure 6.6.

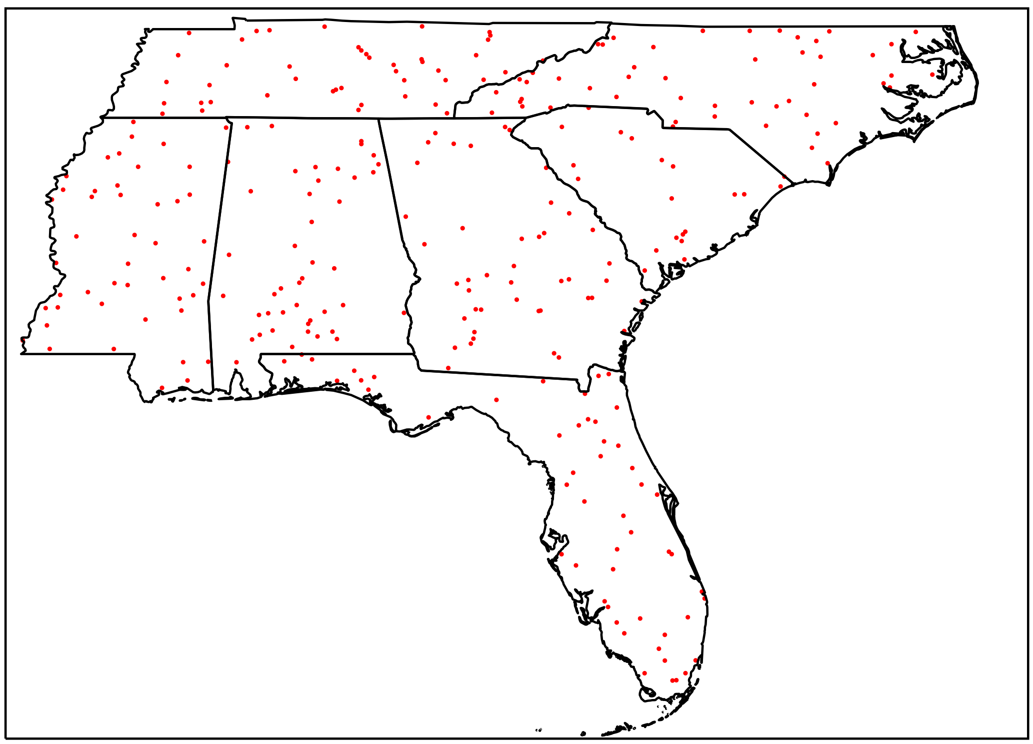

Figure 6.6: CyanoTracker Uniformly distributed POIs

Considering that locations relevant to the CyanoTracker project are not necessarily uniformly distributed through space, and rather may have several near one body of water, a second data set was created by finding 100 randomly distributed points. These points were then used as cluster centers and used to locate 3 points in that nearby vicinity. In this way small areas of cyanobacteria activity that are in close proximity may be simulated. The results of this point generation may be seen in Figure 6.7.

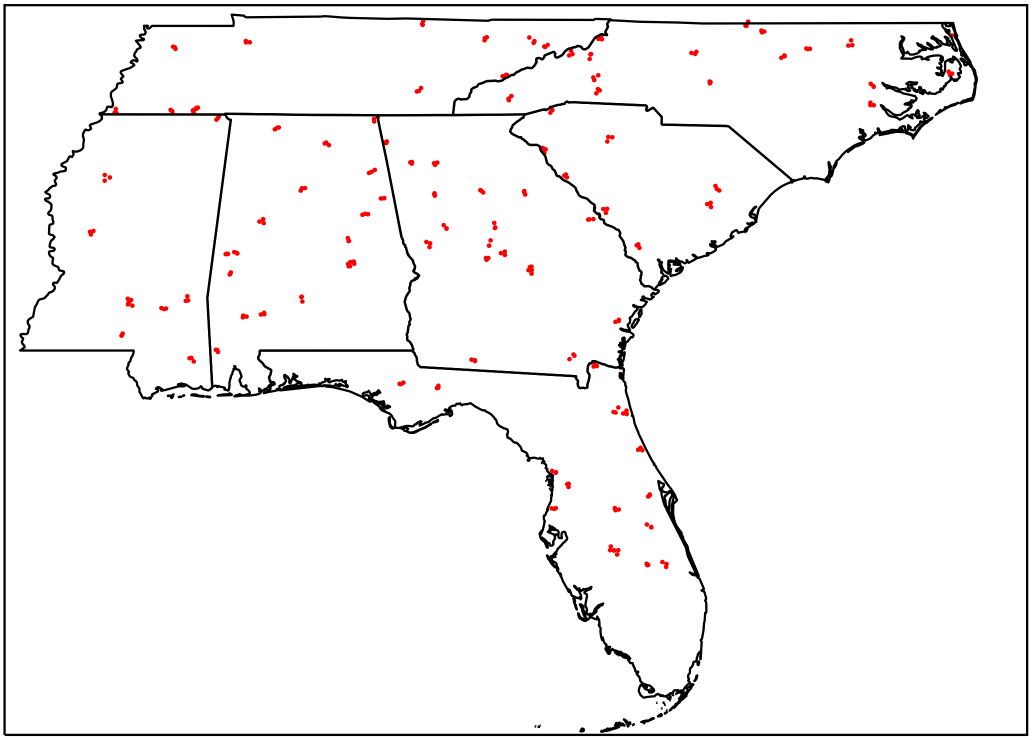

Figure 6.7: CyanoTracker Clustered POIs

A set of 300 simulated smartphone locations was also created for use in both experiments. These device locations were treated as uniformly distributed throughout the same geographic area. The result of the device location distributed can be seen in Figure 6.8. The locations for the three sets were all chosen at random using the QGIS desktop software version 2.8.6-Wien.

### 6.1.4 Results

The results of the experiment show that the overhead increases as the privacy radius is increased. This overhead is also increased with the use of clustered points-of-interest.

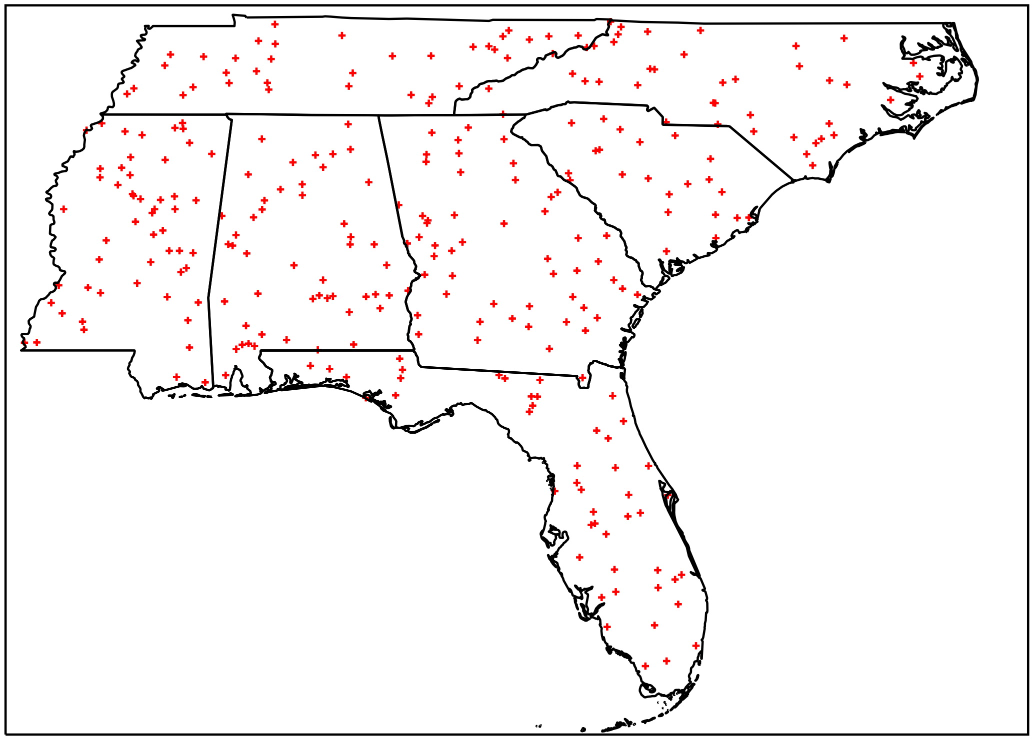

Figure 6.8: CyanoTracker Device Locations

Figure 6.9 depicts the results measured for the 10-mile travel radius configuration with both clustered and uniformly distributed points of interest. We find that there is a small increase in the overhead for the clustered POI configuration.

The results for the 30-mile travel radius are depicted in Figure 6.10. We find a slightly more pronounced increase in both the overhead difference incurred with the clustered POIs as well as with the total overhead compared to the smaller travel radius.

Figure 6.11 we find an even more pronounced overhead difference between the clustered and uniformly distributed POIs.

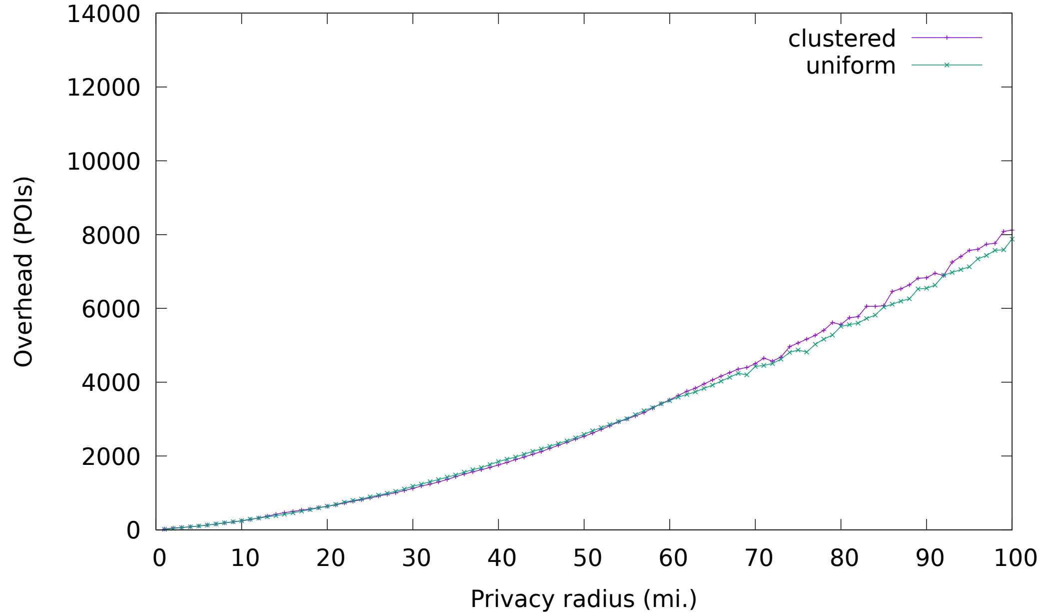


Figure 6.9: CyanoTracker 300 device, 300 poi, travel radius 10 mi.

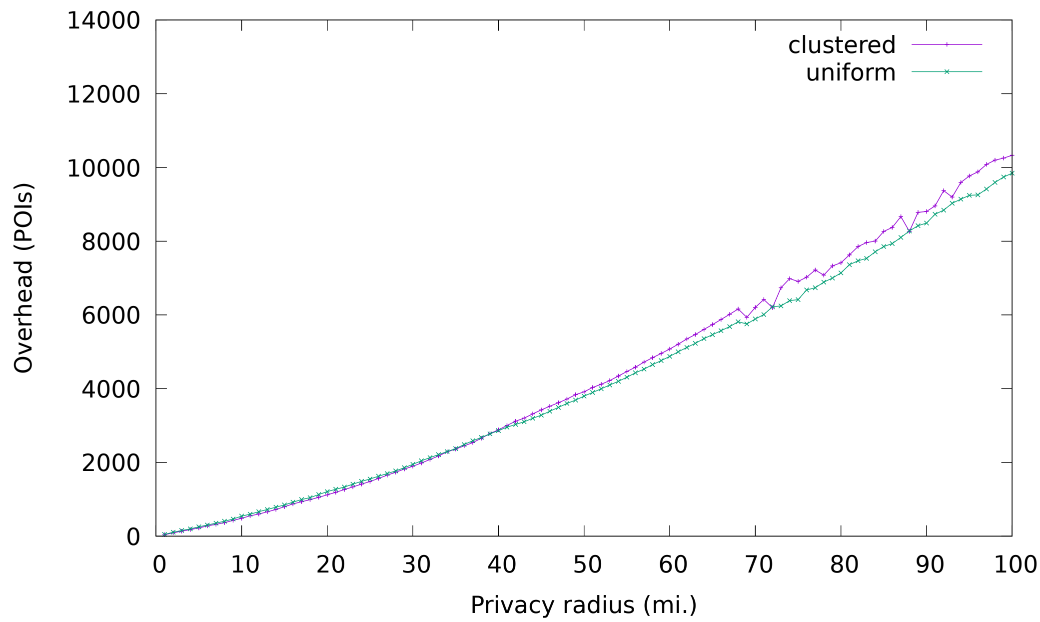


Figure 6.10: CyanoTracker 300 device, 300 poi, travel radius 30 mi.

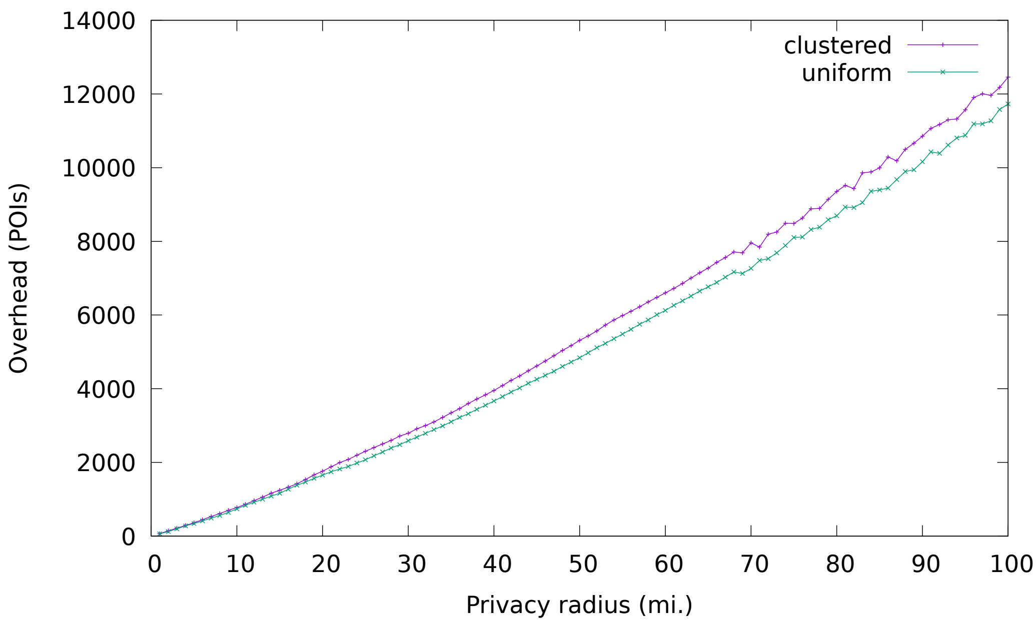


Figure 6.11: CyanoTracker 300 device, 300 poi, travel radius 50 mi.

# Chapter 7

## Experimental Evaluation

### 7.1 CyanoTracker

In order to evaluate this reporting system, an experiment has been conducted to measure the time, bandwidth, and estimates of the energy requirements associated with the reporting process. Six test report configurations have been used (see Table 7.1).

Tests were done using both Wi-Fi and cellular network access. Measurements for both energy and time were taken with varying degrees of signal strength and quality. Table 7.2 lists the test reporting locations used in these experiments.

All tests have been performed using the Android reporting application. The test device used was a Samsung Galaxy S6 running Android 6.0. In order to facilitate making energy and time measurements, customized version of the application was

Table 7.1: CyanoTracker Test Reports

| Config | Description | File Size |
|---|---|---|
| 1 | Blank form, no image | 0 |
| 2 | Default form, no image | 0 |
| 3 | Default form, tiny image | 1 MiB |
| 4 | Default form, small image | 2 MiB |
| 5 | Default form, medium image | 4 MiB |
| 6 | Default form, large image | 8 MiB |

Table 7.2: CyanoTracker Test Locations

| Location | Description | Signal |
|---|---|---|
| A | WiFi, strong | -45 dBM |
| B | WiFi, medium | -64 dBm |
| C | WiFi, weak | -76 dBm |
| D | Cellular 4G LTE | -93 dBm |

made which dit not require interaction with the user interface. This version was created to repeat each test ten times and to report the total time spent or energy consumption estimate.

In order to determine the bandwidth requirements, the six test reports were filed from the mobile device while running a packet sniffer. The total size of transmitted data was recorded at the network level to account for protocol overhead. The bandwidth requirement effects both the time and energy. The bandwidth consumption of the application appears to be almost entirely dependent upon the presence (and size) of any image attachments. These results are depicted in Figure 7.1. This is not very surprising considering the text report fields are simply UTF-8 characters. The app limits the total length of these fields to be less than 1,024 characters. These elements therefore contribute up to 1 kilobyte, whereas a 4-megabyte image will be 4,000 times larger.

The estimated energy requirements of the application were measured on the mobile device to consider the impact on the battery life. In order to make these measurements an application known as the PowerTutor monitor was run during the report filing process. The effect of report size on energy consumption was measured for the test report configuration. These results are depicted in Figure 7.2.

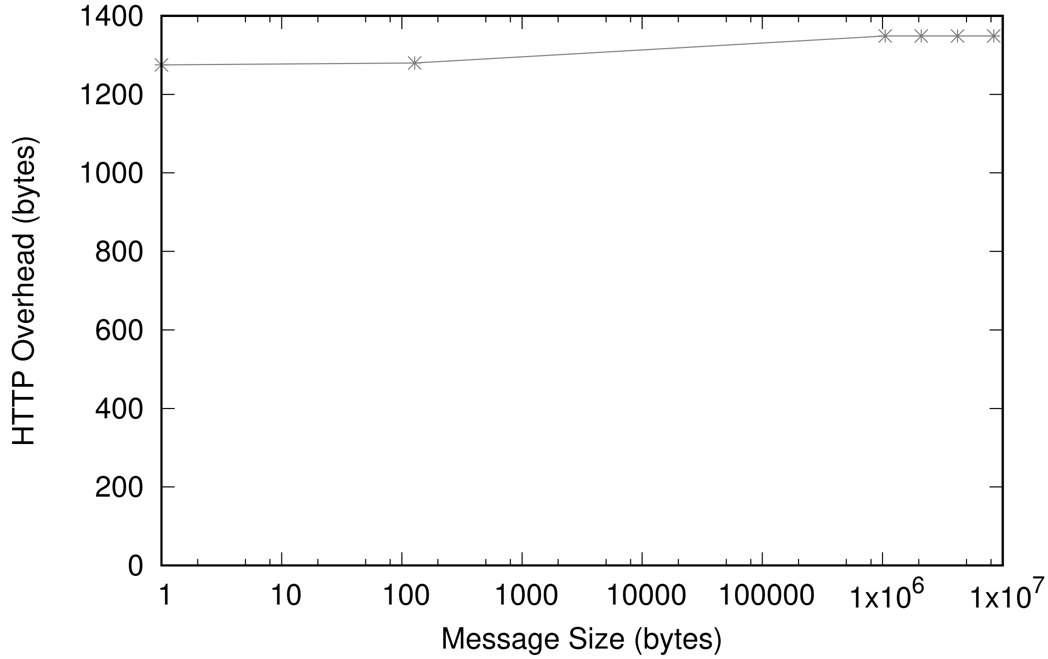


Figure 7.1: Bytes overhead by report size

These measurements were taken for each of the test report locations. Ten samples were taken each time and averaged. The results are displayed in Figure 7.3. These results indicate that having a strong connection reduces the energy consumption associated with the report filing process. Wi-Fi appears to provide a significant advantage. This strengthens the argument for delayed report submission during periods of poor network connectivity or low battery life.

The time to report has been extracted from the traces collected previously. Each value represents the average of ten values recorded. The results are found in Figure 7.4. This similarly shows that the time to submit is improved greatly by on stronger network connections. As the user should not be expected to actively

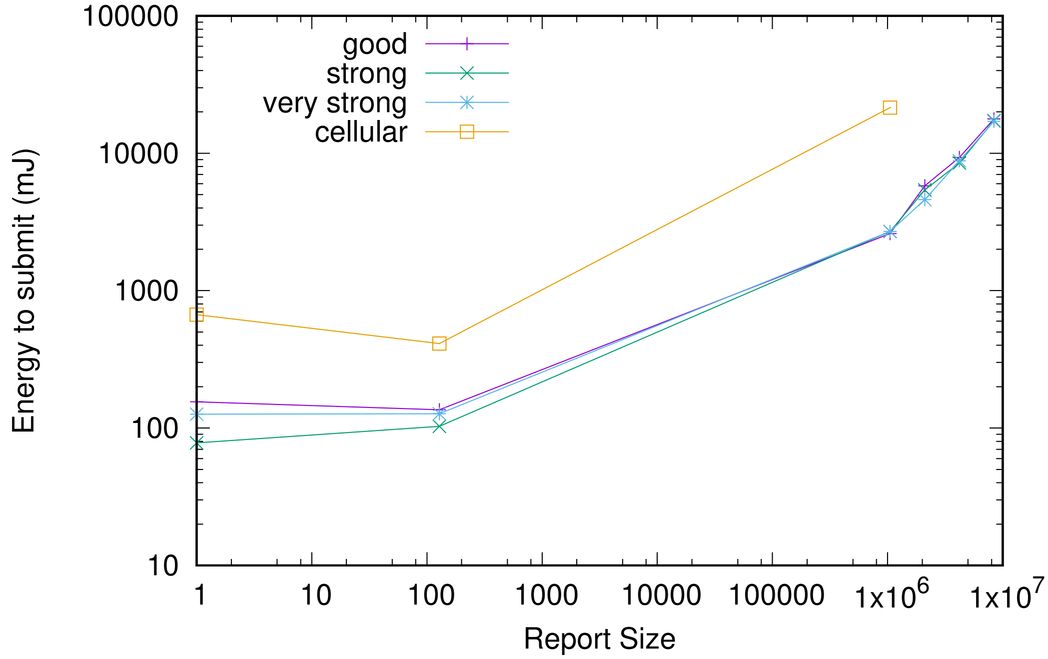


Figure 7.2: Energy consumed by connection, report size

wait on the device for report submission, queued and background submission is a desirable alternative.

## 7.2 PillSafe

In order to evaluate the PillSafe system, several experiments were performed to measure the response time and bandwidth consumption. Additionally, the decomposition of response time into the network transfer time and the actual execution time has been considered. Figure 7.5 shows the response time measured over three different network configurations for three different requested products. In each case, the average of 10 requests was taken. In all three cases the response time over the

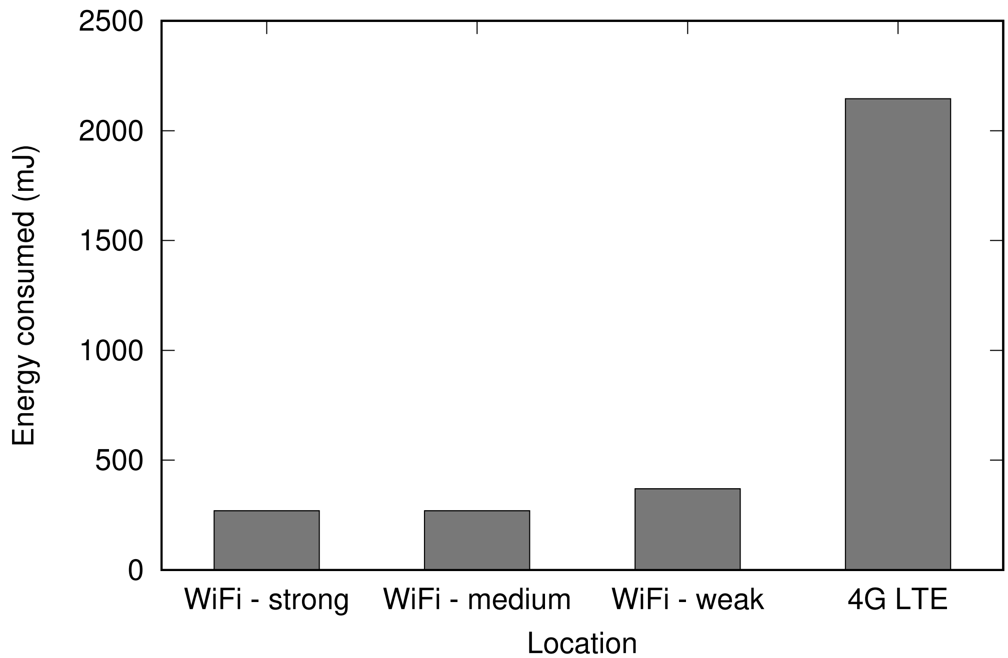


Figure 7.3: Energy consumed by location – 1 MiB report

cellular network is several times larger than that of the ethernet or Wi-Fi connections. Even over a cellular connection, however, the response time was adequate for the application.

The bandwidth requirements to satisfy requests were measured by capturing packets between the client and server during the processing of a request. These traces were then analyzed to determine the amount of network bandwidth used. The results of this experiment are shown in Figure 7.6. It is clearly seen that the state transfer methodology incurs a significant overhead compared to centralized state that does not require as much network bandwidth. In this case it is the larger profile and administration history of user 2 that leads to the increased overhead.

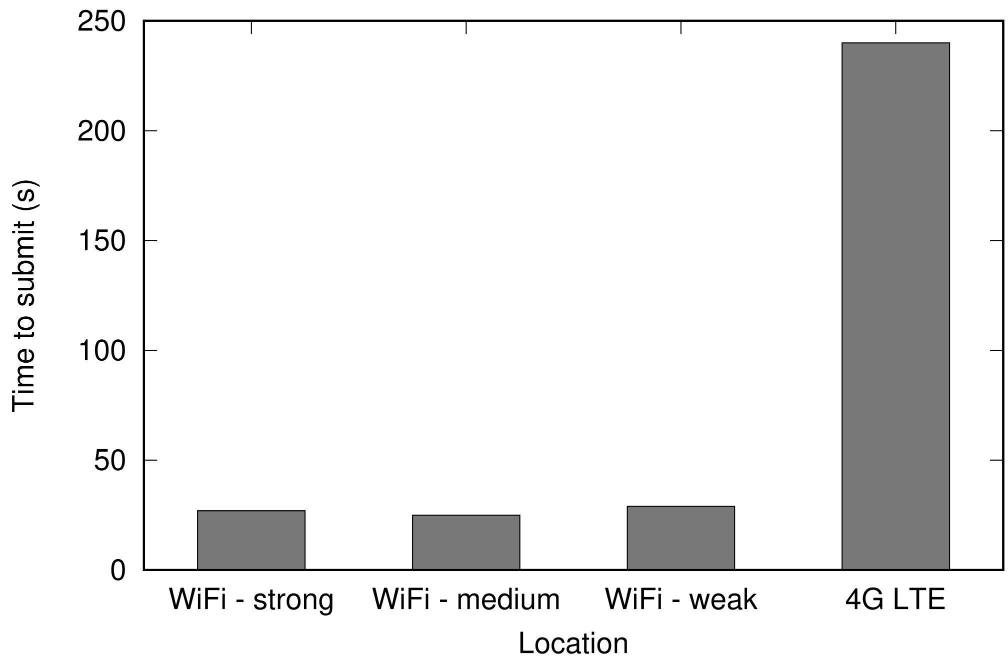


Figure 7.4: Time to submit by location

The response time as it relates to a user and the chosen product was measured and recorded. These measurements were taken to determine how response times would be affected by the growing size of the user profile and the chosen drug product. The results of this experiment are depicted in Figure 7.7. The results show that the response times are not heavily related to the size of the profile and are instead more likely the result of overhead.

An additional test was performed to determine the response time behavior as the record length grows substantially. This was done with both the centralized state methodology as well as with state transfer. The results of this experiment are found in Figure 7.8. The results show that the centralized state method remains feasible for longer.

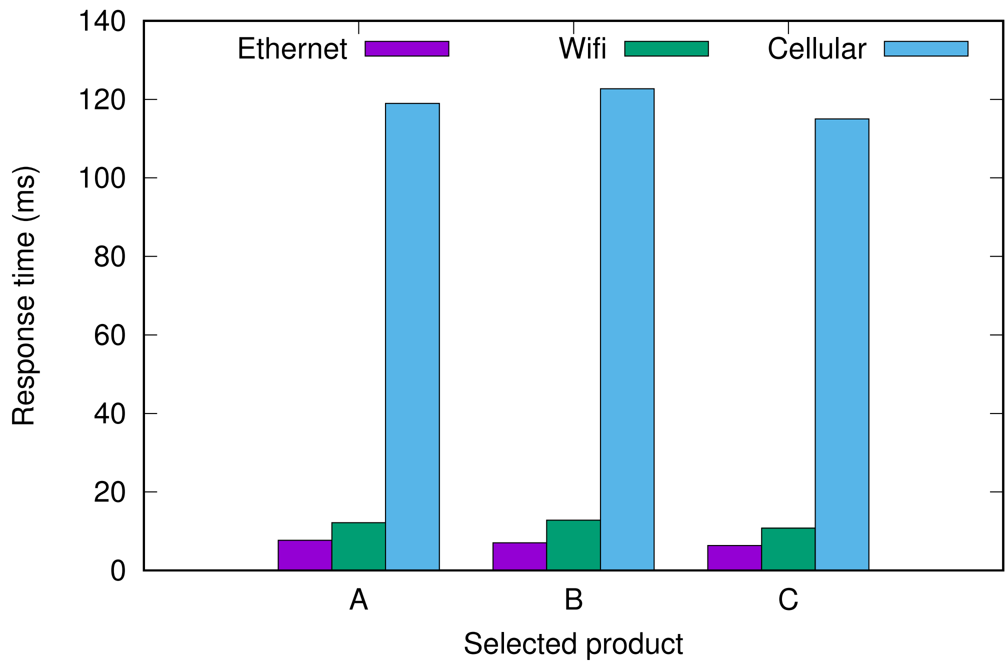


Figure 7.5: Response time by access method

Response times are composed of both an actual execution time as well as the time spent doing the network transfer. This proportion is depicted in Figure 7.9. It can readily be seen that a large portion of the response time is spent doing the network transmission.

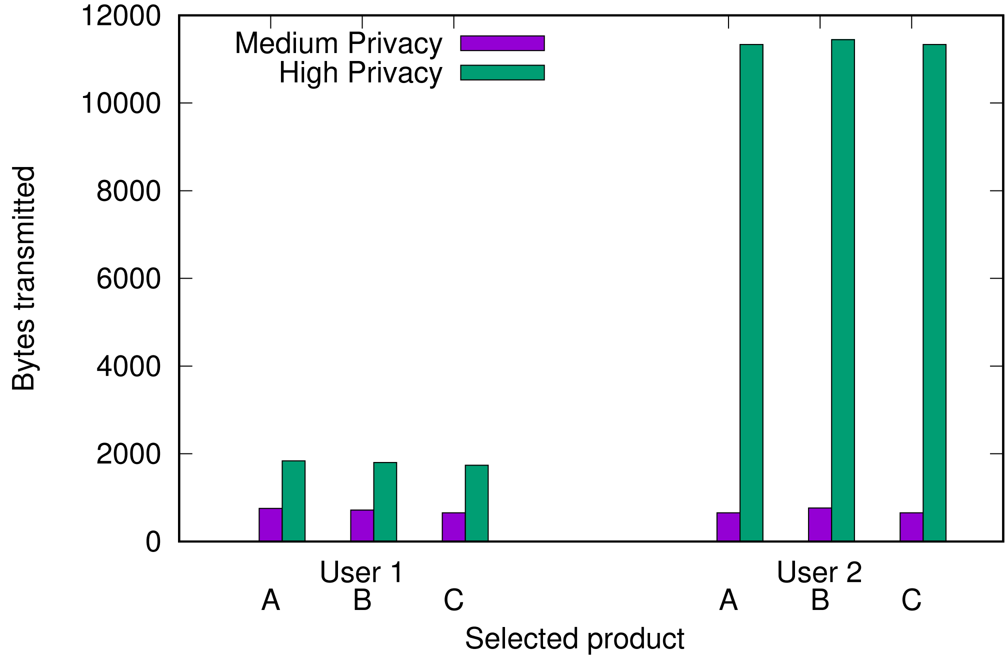


Figure 7.6: Total bandwidth to handle request

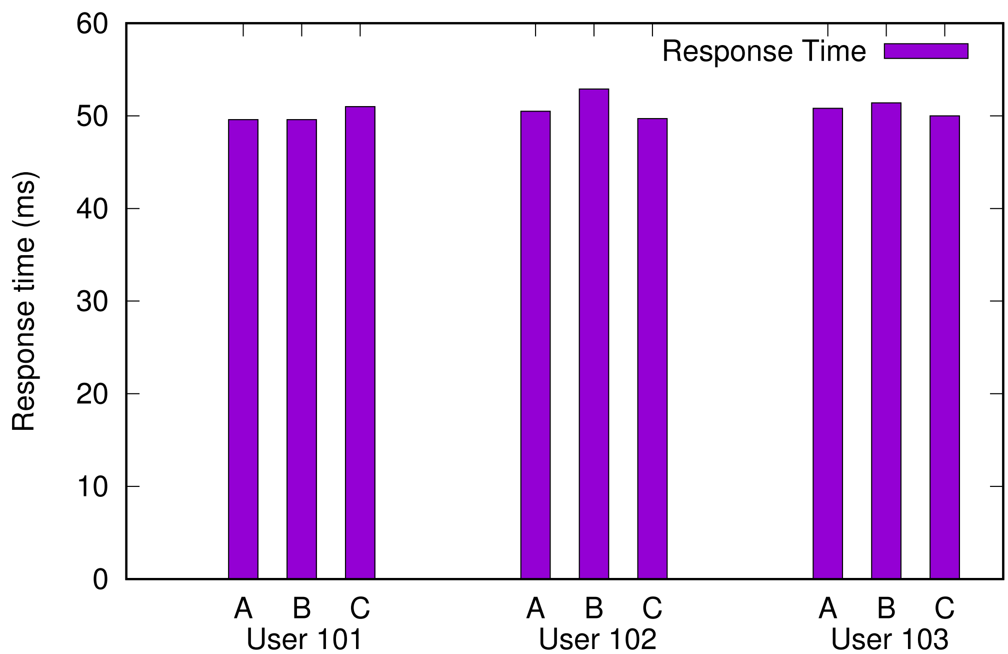


Figure 7.7: Response time by user, product

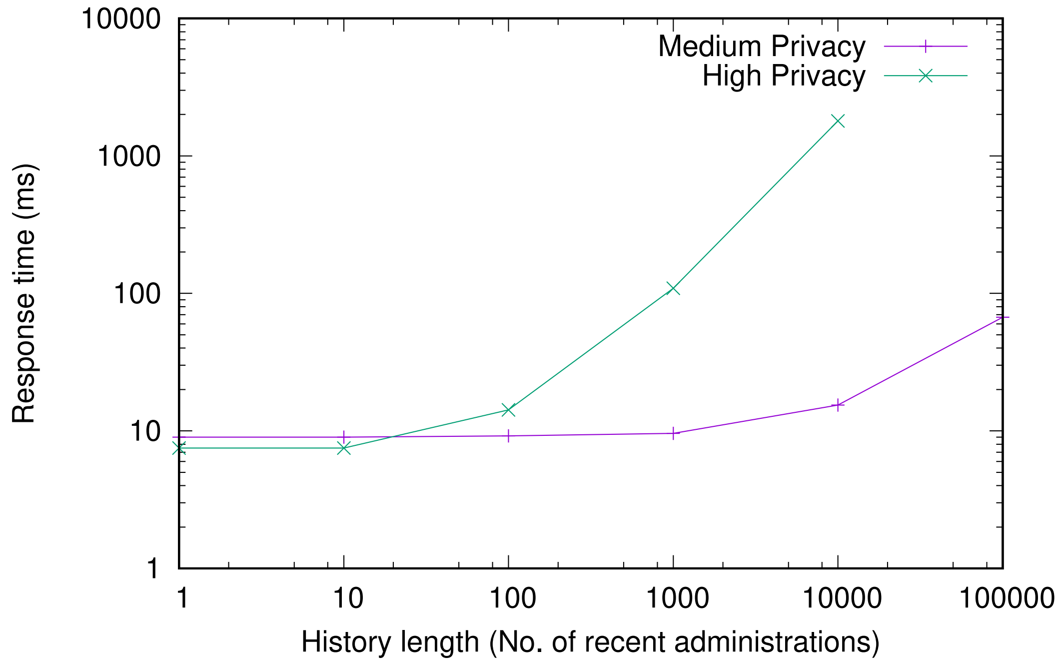


Figure 7.8: Response time by history length, wired network

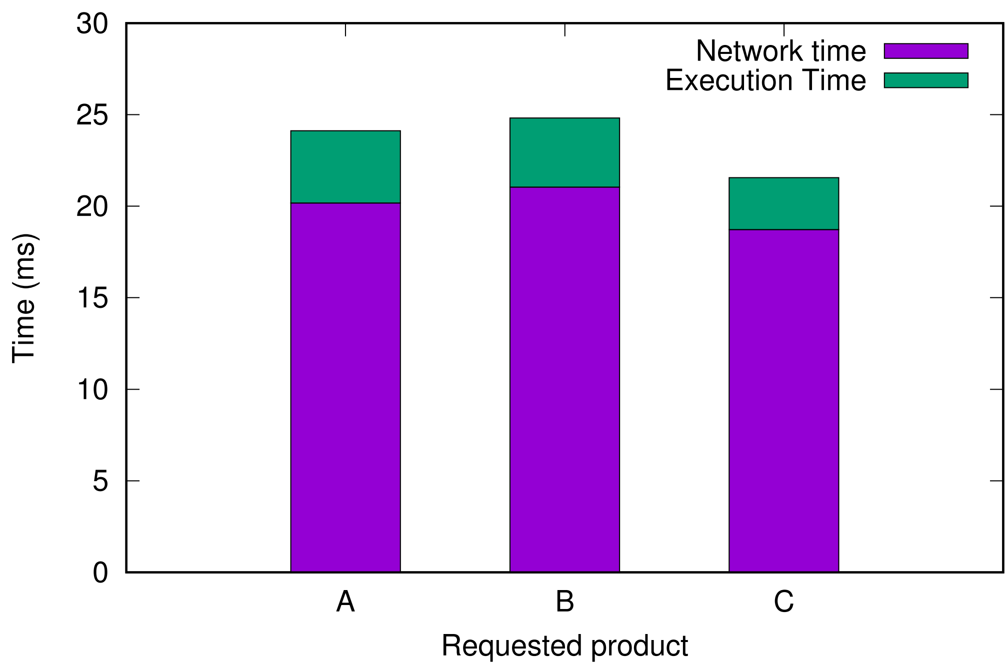


Figure 7.9: Response time – execution time and network times

# Chapter 8

## Conclusion & Future work

In this work three key challenges facing mobile applications for environmental and health monitoring have been investigated. These challenges considered the context of interactions with citizen science applications, the motivations of users and potential incentivization strategies, and provisions to help accommodate privacy-conscious users and enable their participation. This work provides a conceptual framework for understanding these concepts and explores their application through two case-study projects. This framework describes important considerations to be made by citizen science application designers.

The capture and use of contextual information related to reports allow citizen science applications to simplify the reporting process by automatically capturing relevant data. This reduces the burden on the user and increases the data collected by the project. Application designers should identify key elements of context and determine how best to integrate them into the reporting process.

Adequate consideration of user motivation, report valuation, and potential incentivization schemes promote user interaction and seek to additionally improve the quality of submitted data. Application designers must identify these motivations in a project-by-project basis and establish incentivization schemes which reward users based on the value of their contributions to the project.

The creation of adequate privacy controls permits the participation of users who might otherwise be reluctant to engage with a project. These controls must strike a balance between privacy and the quality of data reported. Application designers must determine to what degree the accuracy of various aspects of the report is important and how a tradeoff can be made to improve the privacy of users. These controls are ideally flexible so that those who demand more privacy may be appeased while less concerned users may contribute more openly.

Future work in the area ought explore user motivation and incentivization in greater detail, surveying users from more diverse backgrounds and exploring additional report valuation mechanisms. Additional means of trading off privacy and report detail should be explored to improve the flexibility of these controls and to soften this gradient.

Through the completion of this work it is found that effective management of interaction context, user incentivization, and optional privacy controls benefit participatory sensing mobile applications for environmental and health monitoring by improving the quality and quantity of data shared.

## 8.1 Future work

### 8.1.1 CyanoTracker

There are at least two useful remaining additions to the CyanoTracker application. These include an enhancement to notify users of nearby points of interest based more appropriately on the distance that must be traveled via roads. The second is to improve the context-awareness of the application to determine the mode of transportation of the user to more effectively determine the proximity of these points of interest. For example if the user is in a boat the ease of accessing a location may be greater than if the user was in a car on a nearby road.

#### Nearby location enhancements

The nearby location feature of the CyanoTracker app is limited by the fact that it only considers straight-line distance between two points. Users, however, are typi-

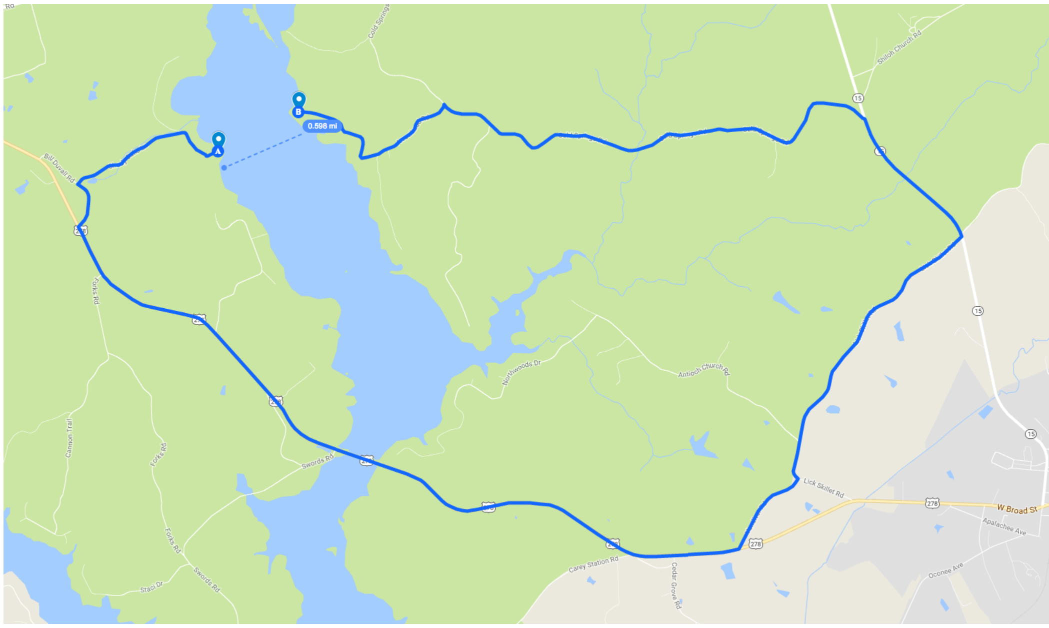


Figure 8.1: CyanoTracker Nearby Location

cally going to be required to travel by roads or paths or may be blocked by geographical features. As a result it may take considerably longer to arrive at a location than it might originally be thought by straight-line distance. In this way the application is currently considering what might be the best-case scenario for travel to a location of interest.

A more realistic estimate may be provided by use of adequate trip planning software. There are already APIs available that can facilitate resolving GPS coordinates to nearby street addresses. These addresses could then be used to query a trip planning service like the Google Maps API to determine how far by automobile a location of interest may be. This will help to ensure that we do not falsely suggest locations that are actually much further away than expected after considering the reality of driving to a location.

#### Context-aware behavior

An improvement to the context-aware behavior mechanism that builds on the previously mentioned enhancements to nearby location is the ability to notice whether a user might be in a boat on a body of water. If this is the case then the time and distance requirements to travel to a region of interest may be dramatically different. By careful examination of GPS coordinates and the nearby roads and geography we may try to determine whether the user is on the water.

### 8.1.2 PillSafe

A useful extension to the PillSafe service is to enhance the system to allow for the reporting of side effects. This provides additional useful information for researchers. In this way the researchers would also have a rich history of the user medication administrations as well as the patient health record, as well as timestamps of the potential causes and effects. This adds an additional citizen-science component to the PillSafe application.

In this case the basic interaction with the system is to say “I’m experiencing side effect X.” The additional context of the interaction is the administration history and the patient profile. One question to consider is how much of the history should be included. The answer to this may depend on issues such as the medication(s) in question and features of the EHR.

# Bibliography


[1] About SETI@home. https://setiathome.berkeley.edu/sah_about.php. Accessed: 2017-10-11.

[2] Great Internet Mersenne Prime Search - PrimeNet. https://www.mersenne.org. Accessed: 2017-10-11.

[3] History of the Christmas Bird Count | Audubon. http://www.audubon.org/history-christmas-bird-count. Accessed: 2017-10-11.

[4] Marine Debris Tracker. www.marinedebris.engr.uga.edu. Accessed: 2017-10-11.

[5] MERIS - Earth Online - ESA. http://earth.esa.int/web/guest/missions/esa-operational-eo-missions/envisat/instruments/meris. Accessed: 2017-10-10.

[6] RICK BONNEY, CAREN B COOPER, JANIS DICKINSON, STEVE KELLING, TINA PHILLIPS, KENNETH V ROSENBERG, and JENNIFER SHIRK. Citizen science: A developing tool for expanding science knowledge and scientific literacy. *BioScience*, 59(11), 2009.

[7] Andrew T Campbell, Shane B Eisenman, Nicholas D Lane, Emiliano Miluzzo, Ronald A Peterson, Hong Lu, Xiao Zheng, Mirco Musolesi, Kristóf Fodor, and Gahng-Seop Ahn. The rise of people-centric sensing. *IEEE Internet Computing*, 12(4):12–21, 2008.

[8] Daniel E Canfield Jr, Claude D Brown, Roger W Bachmann, and Mark V Hoyer. Volunteer lake monitoring: testing the reliability of data collected by the florida lakewatch program. *Lake and Reservoir Management*, 18(1):1–9, 2002.

[9] Wayne W Carmichael. Health effects of toxin-producing cyanobacteria:the cyanohabs. *Human and ecological risk assessment: An International Journal*, 7(5):1393–1407, 2001.

[10] Wayne W Carmichael, SM Azevedo, Ji Si An, RJ Molica, Elise M Jochimsen, Sharon Lau, Kenneth L Rinehart, Glen R Shaw, and Geoff K Eaglesham. Human fatalities from cyanobacteria: chemical and biological evidence for cyanotoxins. *Environmental health perspectives*, 109(7):663, 2001.

[11] Ni-Bin Chang, Benjamin Vannah, and Y Jeffrey Yang. Comparative sensor fusion between hyperspectral and multispectral satellite sensors for monitoring microcystin distribution in lake erie. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 7(6):2426–2442, 2014.

[12] Ingrid Chorus and Jamie Bartram. Toxic cyanobacteria in water: a guide to their public health consequences, monitoring and management. 1999.

[13] CHPA Statistics on OTC use. `http://www.chpa.org/MarketStats.aspx`.

[14] Jeffrey P Cohn. Citizen science: Can volunteers do real research? *AIBS Bulletin*, 58(3):192–197, 2008.

[15] CVS Check for Drug Interactions. `http://www.cvs.com/drug/interaction-checker.jsp`.

[16] Dailymed. `http://dailymed.nlm.nih.gov/dailymed/index.cfm`.

[17] Moumita Chanda Das. Gamification in CyanoTracker : Motivate Users to Observe Cyanobacterial Harmful Algal Bloom. Master's thesis, University of Georgia, May 2017.

[18] John A Downing, Susan B Watson, and Edward McCauley. Predicting cyanobacteria dominance in lakes. *Canadian journal of fisheries and aquatic sciences*, 58(10):1905–1908, 2001.

[19] Drugs.com Interaction Checker. `http://www.drugs.com/drug_interactions.php`.

[20] Matthew Dunbabin and Lino Marques. Robots for environmental monitoring: Significant advancements and applications. *IEEE Robotics & Automation Magazine*, 19(1):24–39, 2012.

[21] Lea S. Eiland, Maria L. Salazar, and Thomas M. English. Caregivers' Perspectives When Evaluating Nonprescription Medication Utilization in Children. *Clinical Pediatrics*, 2008.

[22] Epocrates Interaction Checker. `http://www.epocrates.com/`.

[23] Jakob Eriksson, Lewis Girod, Bret Hull, Ryan Newton, Samuel Madden, and Hari Balakrishnan. The pothole patrol: using a mobile sensor network for road surface monitoring. In *Proceedings of the 6th international conference on Mobile systems, applications, and services*, pages 29–39. ACM, 2008.

[24] Philip Aspden .. [et al.], editor. *Preventing Medication Errors.* National Academies Press, 2007.

[25] FDA Consumer Updates: Have a Baby or Young Child With a Cold? Most Don't Need Medicines. `http://www.fda.gov/ForConsumers/ConsumerUpdates/ucm422465.htm`.

[26] Fda consumer updates: Fda 101: Medication errors. `http://www.fda.gov/ForConsumers/ConsumerUpdates/ucm048644.htm`.

[27] Medication Errors. `http://www.fda.gov/Drugs/DrugSafety/MedicationErrors/`.

[28] FDA: Now Available Without a Prescription. `http://www.fda.gov/Drugs/ResourcesForYou/Consumers/ucm143547.htm`.

[29] Jennifer L Graham, Keith A Loftin, and Neil Kamman. Monitoring recreational freshwaters. *Lakelines*, 29:18–24, 2009.

[30] Lee M. Hampton, Duc B. Nguyen, Jonathan R. Edwards, and Daniel S. Budnitz. Cough and cold medication adverse events after market withdrawal and labeling revision. *Pediatrics*, 132(6):1047–1054, 2013.

[31] Karl E Havens. Cyanobacteria blooms: effects on aquatic ecosystems. In *Cyanobacterial harmful algal blooms: state of the science and research needs*, pages 733–747. Springer, 2008.

[32] Jeffrey B Hecht. Using a pda for field data collection. 1997.

[33] Gregory Hitz, François Pomerleau, Marie-Eve Garneau, Cédric Pradalier, Thomas Posch, Jakob Pernthaler, and Ronald Y Siegwart. Autonomous inland water monitoring: Design and application of a surface vessel. *IEEE Robotics & Automation Magazine*, 19(1):62–72, 2012.

[34] Jenna R Jambeck and Kyle Johnsen. Citizen-based litter and marine debris data collection and mapping. *Computing in Science & Engineering*, 17(4):20–26, 2015.

[35] Mati Kahru. Using satelites to monitor large-scale environmental changes: a case study of cyanobacterial blooms in the baltic sea. *Monitoring algal blooms:*

*new techniques for detecting large-scale environmental changes, Springer-Verlag, Heidelberg Berlin*, pages 43–61, 1997.

[36] Tiit Kutser, Liisa Metsamaa, Niklas Strömbeck, and Ele Vahtmäe. Monitoring cyanobacterial blooms by satellite remote sensing. *Estuarine, Coastal and Shelf Science*, 67(1):303–312, 2006.

[37] Nicholas D Lane, Shane B Eisenman, Mirco Musolesi, Emiliano Miluzzo, and Andrew T Campbell. Urban sensing systems: opportunistic or participatory? In *Proceedings of the 9th workshop on Mobile computing systems and applications*, pages 11–16. ACM, 2008.

[38] Understanding acetaminophen. `http://www.tylenolprofessional.com/assets/v4/understanding_acet_2p.pdf`.

[39] N. Lokker et al. Parental Misinterpretations of Over-the-Counter Pediatric Cough and Cold Medication Labels. *Pediatrics*, 2009.

[40] Hideyuki Onodera, Yasukatsu Oshima, Peter Henriksen, and Takeshi Yasumoto. Confirmation of anatoxin-a (s), in the cyanobacterium anabaena lemmermannii, as the cause of bird kills in danish lakes. *Toxicon*, 35(11):1645–1648, 1997.

[41] JM Oneil, Timothy Walter Davis, Michele Astrid Burford, and CJ Gobler. The rise of harmful cyanobacteria blooms: the potential roles of eutrophication and climate change. *Harmful Algae*, 14:313–334, 2012.

[42] Eric Paulos, R Honicky, and Ben Hooker. Citizen science: Enabling participatory urbanism. *Urban Informatics: Community Integration and Implementation*, 2008.

[43] Jenny Preece, Yvonne Rogers, Helen Sharp, David Benyon, Simon Holland, and Tom Carey. *Human-Computer Interaction*. Addison-Wesley Longman Ltd., Essex, UK, UK, 1994.

[44] Kaylan Randolph, Jeff Wilson, Lenore Tedesco, Lin Li, D Lani Pascual, and Emmanuel Soyeux. Hyperspectral remote sensing of cyanobacteria in turbid productive water using optically active pigments, chlorophyll a and phycocyanin. *Remote Sensing of Environment*, 112(11):4009–4019, 2008.

[45] FDA's Safe Use Initiative - Collaborating to Reduce Preventable Harm from Medications, 11 2009.

[46] Save Our Lagoon Project Plan for Brevard County, Florida. `http://loveourlagoon.com/BCsave-our-lagoon-project-plan_final.pdf`.

[47] Laura P. Shone, Jennifer P King, Cindy Doane, Karen M. Wilson, and Michael S. Wolf. Misunderstanding and Potential Unintended Misuse of Acetaminophen Among Adolescents and Young Adults. *Jornal of Health Communication*, 2011.

[48] Jonathan Silvertown. A new dawn for citizen science. *Trends in ecology & evolution*, 24(9):467–471, 2009.

[49] Kelly A Spain, Chad A Phipps, Michael E Rogers, and Barbara S Chaparro. Data collection in the palm of your hand: A case study. *International Journal of Human-Computer Interaction*, 13(2):231–243, 2001.

[50] Hariprasad Trivedi, Akshaya Trivedi, and Mary F. Hannan. Readability and comprehensibility of over-the-counter medication labels. *Renal Failure*, 36(3):473–477, 2014.

[51] WebMD Interaction Checker. `http://www.webmd.com/interaction-checker/`.

[52] Susan B Wilde, Thomas M Murphy, Charlotte P Hope, Sarah K Habrun, Jason Kempton, Anna Birrenkott, Faith Wiley, William W Bowerman, and Alan J Lewitus. Avian vacuolar myelinopathy linked to exotic aquatic plants and a novel cyanobacterial species. *Environmental Toxicology*, 20(3):348–353, 2005.

# Appendix A

## Medications Used in PillSafe

Table A.1: Medications Used in PillSafe

| Drug | Purpose | NDC |
|---:|---|---|
| Acetaminophen | Internal Analgesic | 49035-457-01 |
| Aluminum Hydroxide | Antacid | 11822-0880-7 |
| Aspirin | Internal Analgesic | 49035-370-12 |
| Brompheniramine Maleate | Antihistimine | 52083-681-90 |
| Cetirizine Hydrochloride | Antihistimine | 49035-458-95 |
| Chlorpheniramine Maleate | Antihistimine | 30142-463-62 |
| Dextromethorphan Hydrobromide | Antitussive | 62750-042-10 |
| Diphenhydramine Hydrochloride | Antihistamine / Sleep Aid | 49035-026-20 |
| Docusate Calcium | Laxative | 0363-0257-01 |
| Docusate Sodium | Laxative | 0363-6550-56 |
| Guaifenesin | Expectorant | 0363-0640-08 |
| Ibuprofen | Internal Analgesic | 10202-647-62 |
| Lansoprazole | Acid reducer | 49035-411-01 |
| Loperamide Hydrochloride | Antidiarrheal | 21130-375-22 |
| Magnesium Hydroxide | Antacid / Laxative | 21130-321-71 |
| Meclizine Hydrochloride | Antiemetic | 49035-403-19 |
| Naproxen | Internal Analgesic | 49035-098-27 |

Continued on next page

**Table A.1 – continued from previous page**

| Drug | Purpose | NDC |
|---:|---|---|
| Nicotine Polacrilex | Smoking Cessation | 59779-918-52 |
| Omeprazole | Acid reducer | 0363-0007-03 |
| Phenylephrine Hydrochloride | Nasal Decongestant | 49035-131-68 |
| Pseudoephedrine Hydrochloride | Nasal Decongestant | 0363-0423-21 |
| Ranitidine | Acid reducer | 49035-608-09 |
| Sennosides | Laxative | 49035-348-08 |
| Simethicone | Antiflatulent | 55315-530-60 |

## Appendix B

## Safety Predicates Used in PillSafe

In order to represent the safety precautions found on common over-the-counter medications, the PillSafe system implements the following predicates which operate on user objects. Each item below includes a pseudocode implementation.

**hasCondition**

The purpose of this method is to determine whether the list of conditions on a user profile includes the argument conditionName.

```
boolean hasCondition(conditionName){
    for condition in getConditions():
        if condition.name == conditionName:
            return true
    return false
}
```

**exceedsDrugDoseInWindowHours**

The purpose of this method is to determine whether a user has exceeded a given dosage of a particular drug in a given timeframe (designated in hours).

```
boolean hasCondition(conditionName){
    for condition in getConditions():
        if condition.name == conditionName:
            return true
    return false
}
```

**prescriptionListContains**

The purpose of this method is to determine whether a user's list of medications includes a prescription of a given name.

```
boolean hasCondition(conditionName){
    for condition in getConditions():
        if condition.name == conditionName:
            return true
    return false
}
```

**allergyListContains**

The purpose of this method is to determine whether a user's list of allergies contains the provided argument.

```
boolean allergyListContains(allergyName){
    for allergy in getAllergies():
        if allergy.name == allergyName:
            return true
    return false
}
```

**takingAnyPrescription**

This method determines if there are any entries in the user's prescription drug list

```
boolean takingAnyPrescription(){
    if size(getPrescriptionList()) > 0:
        return true
    return false
}
```

**belowAge**

This method determines if the user is below a provided age.

```
boolean belowAge(theAge){
    if getAge() < theAge:
        return true
    return false
}
```

**aboveAge**

This method determines if the user is above a given age.

```
boolean aboveAge(theAge){
    if getAge() > theAge:
        return true
    return false
}
```

**takingAnyDrug**

This method determines if the user has any drug administrations in the recent history. There is an important problem here related to what time frame we should consider. For the purposes of this project, considering many product labels do not specify a window regarding this rule, we have (semi-arbitrarily) chosen two days.

```
boolean takingAnyDrug(){
    if size(getAdministrationsInTimeFrame(2,
       TimeUnit.DAYS)) > 0:
        return true
    return false
}
```

**isTakingDrug**

This method determines if the user is taking a given drug by examining the recent administration history. There is an important caveat here in that we have chosen the time window for consideration to be two days.

```
boolean isTakingDrug(drugName){
    admins = getAdministrationsInTimeFrame(2,
       TimeUnit.DAYS)
    for(Administration a : admins):
        for(DrugDose dd : getDrugs(a)):
            if dd.getDrug().getName() == drugName:
               return true
    return false
}
```

# Appendix C

## Conditions Used in PillSafe

Table C.1: Conditions Represented in PillSafe

| **Number** | **Condition** |
|---:|---|
| 1 | Asthma |
| 2 | Bloody or black stools |
| 3 | Breast-feeding |
| 4 | Chronic bronchitis |
| 5 | Cough accompanied by excessive phlegm (mucus) |
| 6 | Diabetes |
| 7 | Difficulty in urination due to an enlarged prostate gland |
| 8 | Difficulty in urination due to an enlargement of the prostate gland |
| 9 | Emphysema |
| 10 | Glaucoma |
| 11 | Heartburn |
| 12 | Heart condition |
| 13 | Heart disease |
| 14 | High blood pressure |
| 15 | History of stomach problems |
| 16 | History of stomach problems such as heartburn |
| 17 | Hypertension |

Continued on next page

**Table C.1 – continued from previous page**

| Number | Condition |
|---|---|
| 18 | Irregular heartbeat |
| 19 | Kidney disease |
| 20 | Liver cirrhosis |
| 21 | Liver disease |
| 22 | Magnesium-restricted diet |
| 23 | Pregnancy |
| 24 | Sodium-restricted diet |
| 25 | Stomach ulcer |
| 26 | Thyroid disease |
| 27 | Trouble or pain swallowing food |
| 28 | Trouble urinating due to an enlarged prostate gland |
| 29 | Vomiting with blood |

Appendix D

# CyanoTracker Privacy-preserving POI matching

```
#!/usr/bin/python


import sys


import argparse
import math
from random import random


from geopy.distance import vincenty


latIndex = 0
lngIndex = 1
degreeDistance = 69


def handleargs():
    parser = argparse.ArgumentParser(description =
       "dissertation script -- privacy preserving location
       matching")
    parser.add_argument('deviceFile', action="store",
       type=str, default=None, help="the input device
       location file")
    parser.add_argument('poiFile', action="store", type=str,
       default=None, help="the input poi location file")
```

```
    parser.add_argument('travelRadius', action="store",
       type=float, default=10.0, help="the user's
       willing-to-travel radius")

    return parser.parse_args()

def getObfuscatedLocation(theLocation, privacyRadius):
    angle = random() * 360.0
    distance = random() * (privacyRadius/degreeDistance)

    return (float(theLocation[latIndex])+
            math.sin(math.radians(angle))*distance,
            float(theLocation[lngIndex])+
            math.cos(math.radians(angle))*distance)

def getNearby(allpoints, deviceLocation, radius):
    returnPoints = []
    for point in allpoints:
        dist = vincenty(deviceLocation, point).miles

        if dist <= radius:
            returnPoints.append(point)

    return returnPoints

def privatePoints(allpoints, deviceLocation, travelRadius,
   privacyRadius):
    returnValues = []
```

```
    obfLoc = getObfuscatedLocation(deviceLocation,
       privacyRadius)
    privList = getNearby(allpoints, obfLoc,
       privacyRadius+travelRadius)
    return privList

if __name__ == "__main__":
    pargs = handleargs()
    deviceFileName = pargs.deviceFile
    poiFileName = pargs.poiFile
    travelRadius = pargs.travelRadius

    deviceFile = open(deviceFileName, "r")
    poiFile = open(poiFileName, "r")

    devicesRaw = [tuple(x.strip().split(",")[1::-1])
                  for x in deviceFile.readlines()]

    devices = [tuple([float(z[1]), float(z[0])]) for z in
       devicesRaw]

    pointsRaw = [tuple(x.strip().split(",")[1::-1])
                 for x in poiFile.readlines()]

    points = [tuple([float(z[1]), float(z[0])]) for z in
       pointsRaw]

    for pr in range(1,101):
        privCount = 0
```

```
    travCount = 0
    overCount = 0

    for device in devices:
        priv = getObfuscatedLocation(device, pr)
        privPts = privatePoints(points, device,
           travelRadius, pr)
        travelPts = getNearby(privPts, device,
           travelRadius)
        overCount += len(privPts) - len(travelPts)

    print str(pr) + "\t" + str(overCount)
```