Poster: CrowdZen: Mobile Privacy-Preserving Crowdsourced Data Collection

Joshua Joy
UCLA
jjoy@cs.ucla.edu

Martin Verde
UCLA
martinverde.edu@gmail.com

Keshav Tadimeti
UCLA
ktadimeti@ucla.edu

Tyler Lindberg
UCLA
tlindberg@ucla.edu

Mario Gerla
UCLA
gerla@cs.ucla.edu

I. Introduction

Today, individual users' personal data is being routinely collected for real-time data analytics. As individual users become increasingly concerned about their privacy, the desire of keeping personal data on users' own devices yet still allowing analysts to perform real-time data analytics presents an interesting tradeoff between privacy and utility. Users seek strong *privacy* guarantees, while analysts strive for high-*utility* data analytics with low *latency*, i.e., stream analytics.

CrowdZen demonstrates that meaningful real-time collection and release of aggregate mobile sensor data can be achieved in a privacy-preserving manner. We implement our system on both Android and iOS and describe the design of our private and energy-efficient real-time query response module.

The centralized collection of location data by third parties has created a void depriving data owners of insights into their own personal location data. Straightforward questions such as how crowded a particular point of interest is very difficult to perform real-time estimates.

Naturally, if every person publicly broadcasted their current location every minute it would be trivial to ascertain the crowd levels at particular points of interest. However, clearly this is a privacy concern as data owners would become unnecessarily tracked. The question we ask here is how do we privately collect location data in real-time yet publicly disclose the aggregation information in a useful manner?

This paper presents CrowdZen, the first mobile system which privately collects location data in real-time enabling public disclosure and stream analytics. In CrowdZen, each data owner's personal data resides on the data owner's own device. Once receiving a query, each data owner does not directly respond to the query with the truthful answer. Instead, the data owner locally privatizes their answer based on the randomized response mechanism [3], [1] such that only privatized data is released (rather than the original answer). Randomized response satisfies the local differential privacy requirement such that each data owner's response is independently differentially private, regardless of the amount of differential privacy noise added by other data owners or system components. That is, for a response of ”Yes” the data owner has an equivalent probability of having or not having the sensitive attribute. Thus, randomized response eliminates the need for strong trust assumptions regarding the aggregation mechanism in a distributed setting.

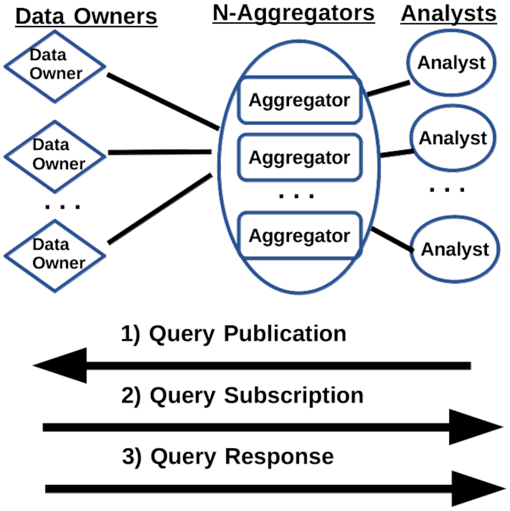


Figure 1: CrowdZen system overview.

In this paper, our contribution is a mobile software which that the first time achieves all of the following for a real-time system:

- privacy-preserving localization module
- energy-efficient query privacy-preserving response module

II. Goals and Problem Statement

We now describe the system goals, performance goals, threat model, and privacy goals of CrowdZen. Figure 1 shows an overview of the flow of queries and responses.

A. System Goals

The system should support analysts who wish to run a population study. The analysts issue a query for those interested data owners that privately and anonymously reply. Analysts are able to formulate long-standing signed queries. These queries continually elicit privatized responses during the defined query epoch. The analysts are deemed to be reputable, e.g., Department of Transportation, National Institutes of Health, or Centers for Disease Control. Each analyst controls an aggregation server.

We use a campus scenario as motivation. Students wish to know the crowd levels at the dining halls, gyms, and libraries in order to avoid long queues and to optimally plan

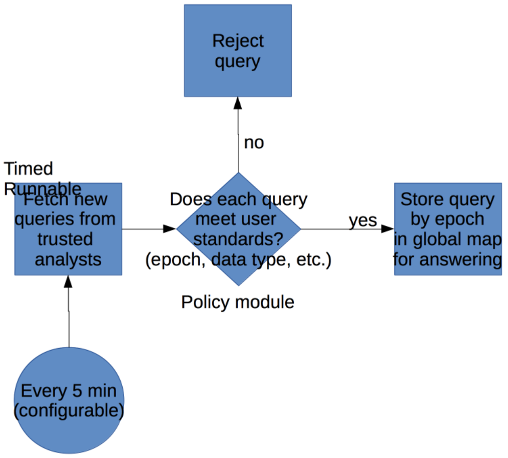

Reject query
no
Timed Runnable
Fetch new queries from trusted analysts
Does each query meet user standards? (epoch, data type, etc.)
yes
Store query by epoch in global map for answering
Policy module
Every 5 min (configurable)

Figure 2: Query fetch.

their schedule. While real-time mobile crowdsourcing has high utility and benefit to society, real-time sensor data harvesting has serious privacy risks. Students (data owners) do not wish to be constantly tracked, as this violates location privacy.

The queries are propagated from a website that students subscribe to. To reduce traffic O/H, or in response to periodic student polls the queries may be posted to an edge website that mobiles of a certain class frequently check (e.g., dining hall website).

The long-standing queries are needed to be fetched only once by each data owner. The responses of data owners and aggregation processing proceeds in epochs. That is every epoch each data owner privately and anonymously transmits their respective answer to the aggregator servers. The aggregation servers then compute the final aggregate using the received responses within this epoch. Epochs are defined on the order of seconds.

B. *Threat Model*

Aggregation servers may try to collude, though we assume there is at least one honest aggregation server. Each aggregation server is owned by a set of distinct reputable analysts.

Aggregation servers are expected to be available and online, so we do not consider denial of service attacks whereby data owners are not able to transmit their responses. We assume aggregation servers are honest-but-curious, i.e., servers do not corrupt the messages though can attempt to read all messages.

C. *Privacy Goals*

We assume all queries are signed and from reputable analysts. This provides provenance in the case of a dishonest analyst that may formulate a specially crafted query that attempts to deprivatize a data owner.

Data owners' privatized location responses should leak no more data than if they were not participating in the population study. Each data owner retains their own data on devices that they control and manage. The data owners then choose to participate in responding to each query. All responses before they leave the data owner are privatized and anonymized. The anonymization mechanism requires only a single honest aggregation server to participate and that there are at least two honest data owners. The privacy mechanism should satisfy the local differential privacy criteria. Thus, there is no centralized or trusted aggregation mechanism that adds differentially private noise. Moreover, neither servers nor data owners can collude to deprivatize the data.

Our goal for anonymity is that a data owner is able to transmit a message such that the message is unable to be linked back to the data owner. That is, a data owner is anonymous within a group of data owners, i.e., the anonymity set. The anonymity scheme should also be robust to traffic analysis. We rely on a public-key infrastructure (PKI) to thwart sybil attacks. However, the use of PKI does not preclude anonymity, as data owners remain anonymous within the anonymity set.

III. Preliminary Results

We would like for a data collection service to run in the background on users' devices in perpetuity to guarantee the maximum utility of collected data. In order for this to happen, the user should ideally not notice an impact on their day-to-day battery life from the service; if they do, they'll likely uninstall CrowdZen. This means the data collection service must be optimized for battery efficiency as much as possible. We accomplish this by listening for sensor data only when the user is within the area of a point of interest and only when the service is about to respond to a query. We also fall back to the last reliable reading if getting an accurate new one takes too long (ex: location), which ensures that the sensors aren't polled excessively. We measured the effectiveness of these optimizations by running the CrowdZen service and the popular location tracking app Waze at the same time on two of the same device model (Nexus 7). Using the Power Tutor app [2] to display the total energy consumption of each app, we measured that CrowdZen used an average of 0.017 Watts, while Waze used an average of 2.177 over a total of 65.5 hours. This is a difference of over two orders of magnitude.

References

[1] Fox, J. A., and Tracy, P. E. *Randomized response: a method for sensitive surveys.* Beverly Hills California Sage Publications, 1986.

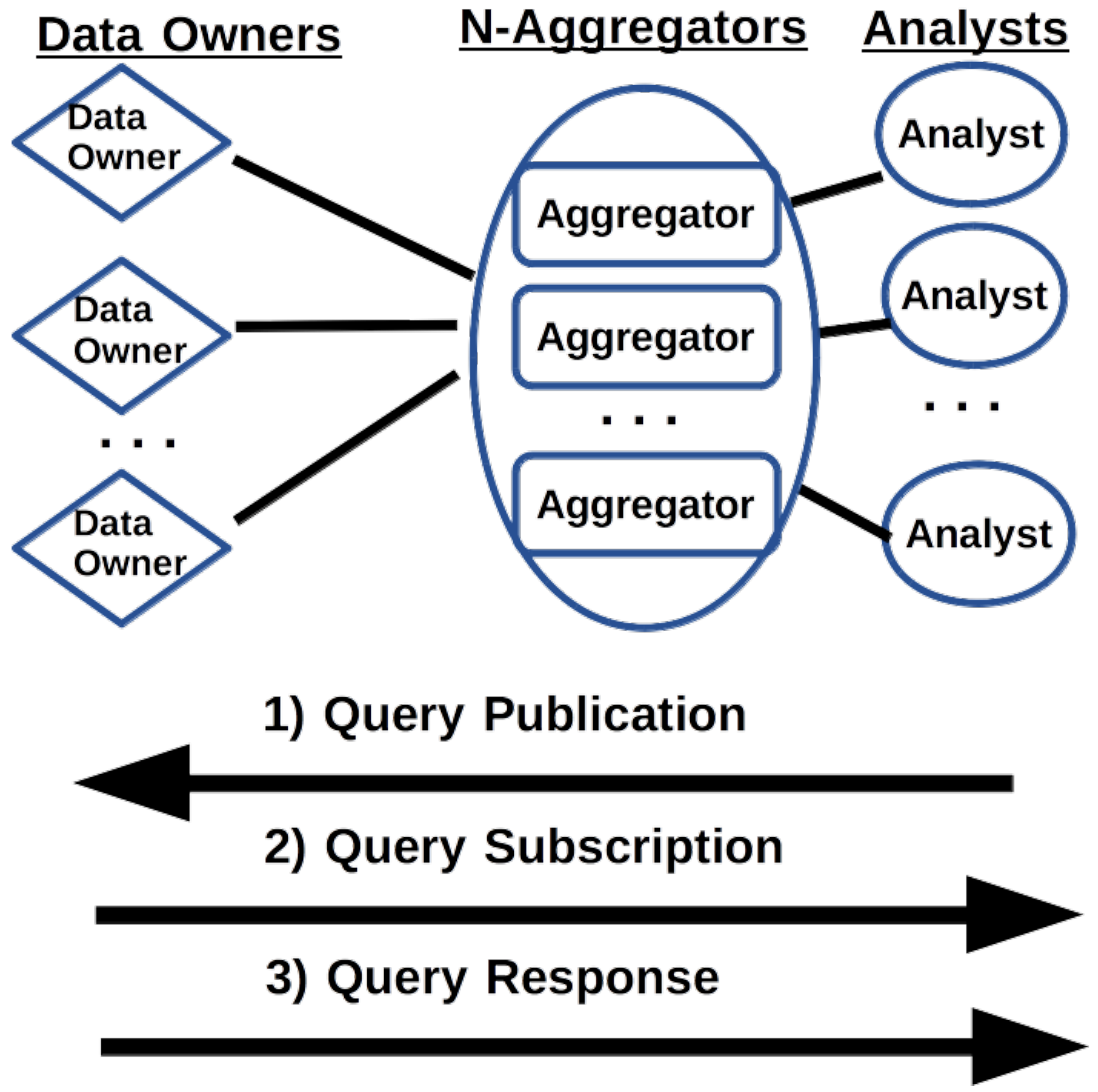
[2] PowerTutor. https://play.google.com/store/apps/details?id=edu.umich.PowerTutor&hl=en.

[3] Warner, S. L. Randomized response: A survey technique for eliminating evasive answer bias. *Journal of the American Statistical Association 60*, 309 (1965), 63–69.

CrowdZen: Mobile Privacy-Preserving Crowdsourced Data Collection

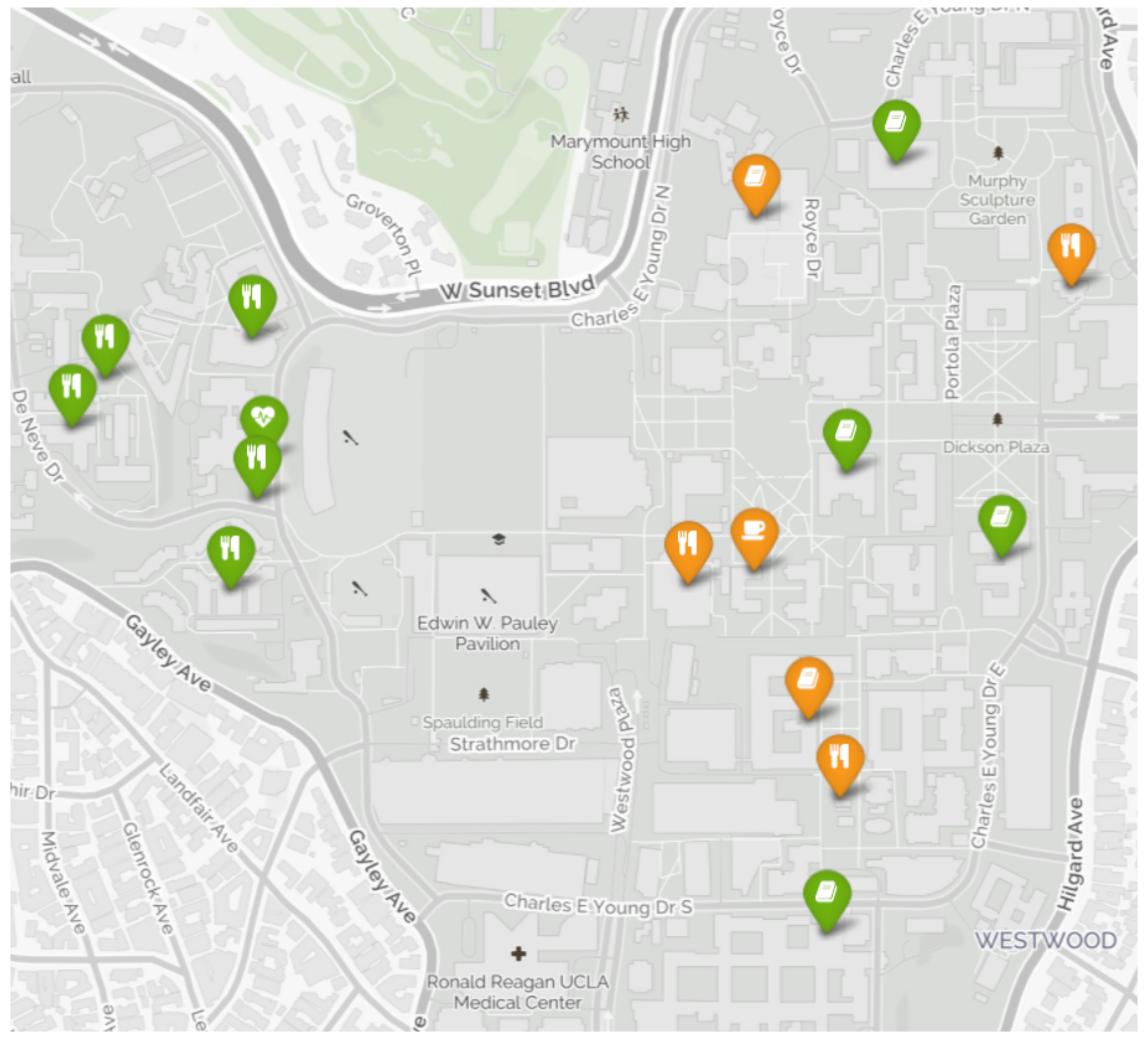
Joshua Joy (jjoy@cs.ucla.edu), **Martin Verde** (martinverde.edu@gmail.com, **Keshav Tadimeti** (ktadimeti@ucla.edu), **Tyler Lindberg** (tlindberg@ucla.edu), **Mario Gerla** (gerla@cs.ucla.edu)
UCLA Computer Science

Problem: Privacy-Preserving Real-Time Data Collection



1. Data owners validate queries before accepting
2. Data owners privately and anonymously write their location

UCLA Map

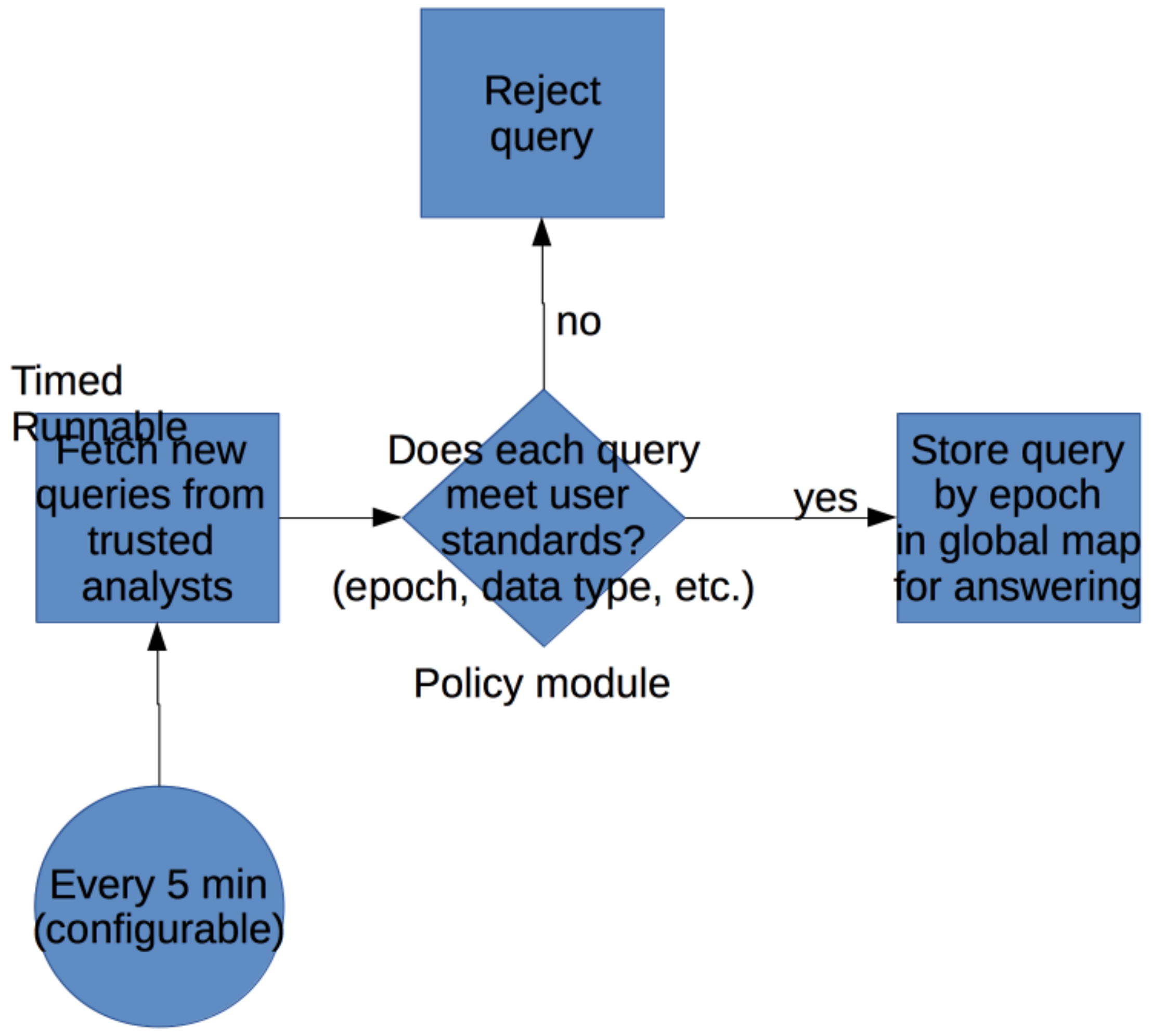


Contributions: Energy-efficient and Privacy-preserving Mobile Apps

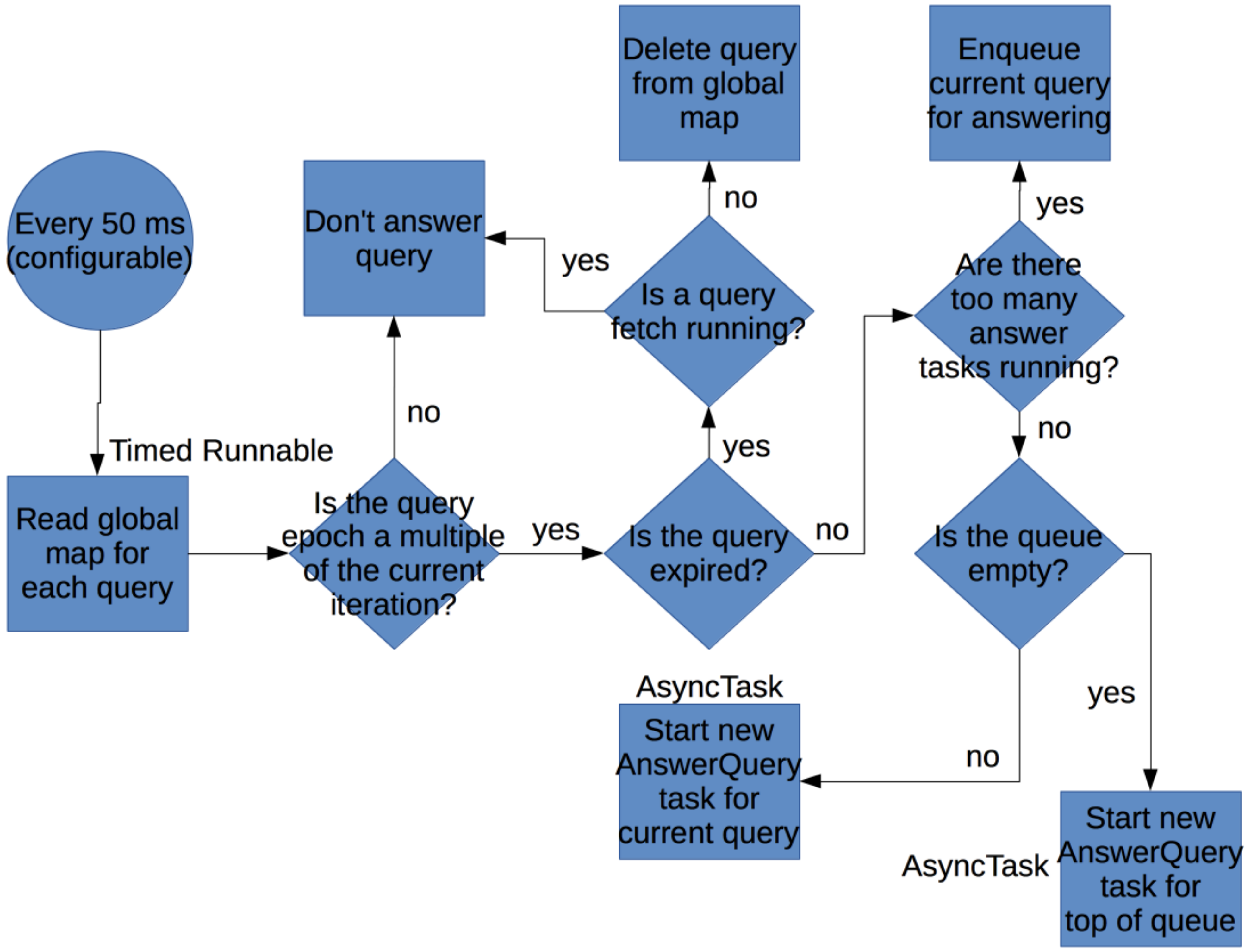
	Waze	Crowdzen
Energy (Watts)	2.177	0.017

Evaluation Period 65.5 hours

Design: Query Fetch


Timed Runnable
Fetch new queries from trusted analysts
Every 5 min (configurable)
Does each query meet user standards? (epoch, data type, etc.)
Policy module
no
Reject query
yes
Store query by epoch in global map for answering

Design: Query Response


Every 50 ms (configurable)
Timed Runnable
Read global map for each query
Is the query epoch a multiple of the current iteration?
no
Don't answer query
yes
Is the query expired?
yes
Is a query fetch running?
yes
no
Delete query from global map
no
Are there too many answer tasks running?
yes
Enqueue current query for answering
no
Is the queue empty?
no
AsyncTask
Start new AnswerQuery task for current query
yes
AsyncTask
Start new AnswerQuery task for top of queue