Network System Challenges in Selective Sharing and Verification for Personal, Social, and Urban-Scale Sensing Applications

Mani Srivastava* Mark Hansen* Jeff Burke*
Andrew Parker* Sasank Reddy* Thomas Schmid* Kevin Chang* Ganeriwal Saurabh‡
Mark Allman† Vern Paxson† Deborah Estrin*

*Center for Embedded Networked Sensing - UCLA †ICSI - Berkeley ‡Google Inc.

ABSTRACT

We envision the blossoming of citizen initiated sensing applications in an urban context, enabled by increasingly affordable and portable sensing hardware, and ubiquitous wireless access to communication infrastructure.

In this paper, we describe the *Partisans* architecture, featuring infrastructure-supported selective data sharing and verification services. This effort represents an evolution of activity on embedded networked sensing from the scientific application space to applications in a space that raises novel issues in privacy, security, and interaction with the Internet.

1 INTRODUCTION

Application context inevitably drives the architecture design choices and the definition of services needed in a network. Over the past decade, the emergence of unanticipated applications of the Internet, such as peer-to-peer file sharing, networked gaming, podcasting, and voice telephony, has contributed to a pressing need to rethink the core Internet infrastructure and its accompanying architectural choices. To truly lay a foundation for tomorrow's infrastructure, however, requires going beyond simply reacting to applications that have already emerged, to proactively considering the architectural implications of new *classes* of applications. A key area in this regard involves embedded sensing technology, presently poised to move beyond scientific, engineering, and industrial domains into broader and more diverse *citizen-initiated sensing in personal, social and urban* settings.

By *sensing*, we mean, "the action of an *automatic device* in detecting, observing, or measuring something."[1] The sensing modalities we are considering include those available on cell phones and handhelds (imagery, video, audio), those typically used in environmental monitoring (temperature, pressure, light-level, etc.), and other modalities supported by mobile sensors.

Today, applications are emerging that draw on sensed information about people, objects, and physical spaces. Sensor-based applications enable new kinds of social exchange: By collecting, processing, sharing, and visualizing sensed information, these applications can offer us new and unexpected views of our communities and environment, such as those depicted in Figure 1. To achieve their potential, these applications require fundamentally new algorithms and software mechanisms. The research described in this paper seeks to identify and develop an overall network fabric architecture that through various services coherently embodies such algorithms and mechanisms.

The applications considered in this paper can be divided into three categories that define how widely sensor data are shared: *Personal, social and urban*, which we sometimes refer to as *PSUS*[2] applications. Medical monitoring is a good example of a *personal application*; observations about a patient's personal space, their heart rate or blood sugar levels, are only shared with the patient's health-care provider. By *social applications*, we mean situations in which data are shared among a group of participants, some, possibly all, of whom contribute data. Applications of this sort are best thought of as combinations of data services and "social networking software." Finally, urban-scale applications involve sharing data with the general public. For audio and imagery, we see precedents in podcasting and in photo sharing services like Flickr[3]. The scope of these applications is much larger and there might be an emphasis on identity control. In some cases participants may prefer to share data anonymously or "pseudonymously."

These emerging applications raise a host of impor-

[1] Oxford English Dictionary

[2] Pronounced as "peices."

[3] http://www.flickr.com

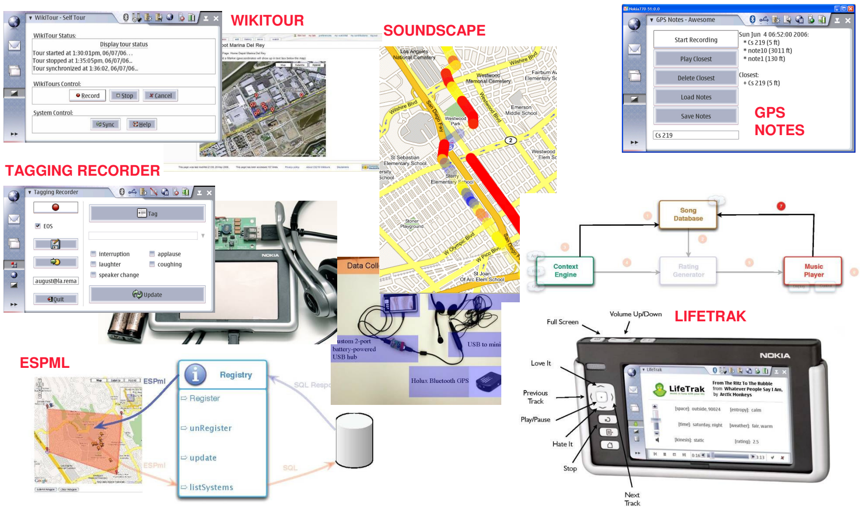


Figure 1: This is a collage of urban sensing applications from CS219 Spring 2006[5]. All of the applications use the Nokia 770 as a platform as well as capture, retrieve, share, and/or interpret sensed information about the user's environment.

tant and challenging questions, whose answers potentially reach deep into the network architecture.

- What mechanisms will enable those who deploy sensors to share data in a controlled way while respecting the privacy of those being sensed?
- Can we assure basic quality checks for data? For example, if a temperature reading is much higher or lower than readings taken from nearby locations, it should be flagged in some manner.
- By providing a suite of services, can we encourage responsible sensing practices?

We will argue that connecting these sensing systems to provide such basic assurances requires new infrastructure that we call "Partisans." Such an infrastructure aims not only to aid organizations in implementing a wealth of proliferating sensing applications, but also to promote citizen-initiated sensing projects.

2 Design Challenges

Data are more valuable if they can be *verified*. For example, a *subscriber* to a data feed might want to know, with some certainty, the time and *specific* location at which a measurement was taken. For some *providers*, such disclosure is too invasive, and they would prefer to only reveal their location in terms of their ZIP code or county. The same kind of resolution control could apply to the time of a measurement, with some data providers choosing only to reveal the hour or day on which data were taken. Naturally, there will always be situations in which data can be shared freely, without restrictions. The emerging network of amateur weather stations is an example of this.[4] No matter what resolution a user is comfortable with, it is important that the context assigned to data be verifiable in some fashion.

By controlling the resolution or context of a measurement, the data contributor is, in effect, defining a privacy policy. We prefer to use the term "selective sharing" because it captures the idea that participants choose the conditions under which their data are divulged. For most of the examples presented so far, the sensing hardware acts in an essentially autonomous way, collecting data at regular intervals or in response to a detected event. Therefore, policies for selective sharing must be implementable as an automated component of a sensing system. Policies should also adapt in response to a contributor's changing public/private context.

It is conceivable that sensors deployed in public places could be expected to provide access to the data they are collecting, but only to people within a certain distance of the device. Here again, we see that decisions about data sharing depend on location and time.

Names touch on how we do dissemination, selective sharing, and verification. The items being named are the data streams published by sensor devices. For example, a mobile phone could have three data streams: an audio stream, a video stream, and a location stream (perhaps something fine grained like GPS, or course grained like reachable cell towers). However, a publicly browsable registry of data streams may disclose information that the publishers wish to keep confidential.

[4] http://www.wunderground.org/

Personal, social, and urban sensing applications, as experienced and consumed by the end user, will straddle both traditional web-based applications, and sensor network applications. We envision a web services architecture that provides a platform to feed data to these applications, in much the same way that applications have sprung up around the Google Maps API and other platforms that give users access to vast amounts of data in a programmatic way over the web. RSS, ATOM and other web feed formats provide a useful, uniform interface to web sites that have stylized update mechanisms. This has enabled the construction of readers, aggregators, and other tools that enable the user to mix, filter, and otherwise experience content in customized ways. In a similar fashion, data streams from sensors should be subject to this kind of end-user manipulation.

2.1 Context Verification and Selective Sharing

Essential to building space-time semantics into the fabric is the ability for it to verifiably measure the location of a node and the time at which it transmits data. The basic measurement part, while not simple under adverse conditions, is conceptually simple and well studied. Using time-stamped message exchange based on protocols such as NTP, a node in the network can measure the clock offset relative to a sensor node [11][6]. Likewise, a base-station in the network can measure distance to a sensor node using radio time-of-flight or signal strength for ranging, and then use multilateration to determine the position of the sensor node [13] using similar distance measurements made at other base-stations. Even simpler would be for the device to measure its own location and time using GPS. However, the key problem is one of verifiability: the location and time estimate must be robust to cheating by a malicious sensor node; manipulation by an external adversary can be guarded against by having the sensor sign its data.

The physical context of sensor data is richer than just location and time. It includes, for example the orientation of the sensor, measurements made by other sensing modalities, and measurements made by other sensors in the vicinity. Clearly, such additional physical context is of utility to the subscriber in interpreting the sensor data or in checking its integrity. For example, the utility of sound level from a directional microphone is significantly increased if the orientation of the microphone is also known. To increase the utility of sensor data, information about the sensor's context can be combined with statistical and physical models of how different sensing modalities are related, and of how measurements made by nearby sensors relate to one another. Indeed, as the sensor infrastructure for PSUS applications proliferates, the increased spatial and temporal density of measurements will inherently provide additional physical context for the validation of a specific sensor measurement. Moreover, application deployments may have self-awareness sensors [8] whose purpose is to acquire information about the physical context as opposed to the phenomenon in which the subscriber might be directly interested.

The context derived from information provided by the sensors themselves is fundamentally different from location and time since the fabric has an independent ability and reason to measure space and time, but has no reason to directly measure the direction of a sensor, temperature in the vicinity of a sensor, etc. The role of the fabric in this case is therefore one of calculating and verifying the context according to application specified rules. For example, an application may request that the fabric corroborate a sensor reading with the readings from nearby sensors by comparing against their average.

The final element in context verification is the ability for the application to exercise control over the context that is revealed to a subscriber. Specifically, the fabric will ensure that even if it has information about the physical context at a finer detail, it does not send to a subscriber more contextual information than what the publisher is willing to share. The reason the fabric has access to higher accuracy data is so that it can better verify and aggregate it. For example, the fabric may know the location of a sensor to within a few meters, but the subscriber may only be willing to share the location information to the ZIP Code level. Likewise, a sensor may be willing to share information only as part of an aggregate in a geographical region. The fabric will deliberately reduce the fidelity of the context information it shares (location, time) or derives from sensor values. In addition, to combat emerging techniques for remote device fingerprinting based on measurements of timestamp drift [9] and localization using latency measurements [7], the fabric may add random jitter to packets. Note that the fabric exercises resolution control only over context; the sensor data values are forwarded unmodified, and it is the publisher's responsibility to blur them a priori as needed.

Besides physical context, also part of a sensor device's network context are network-level identifiers such as host name or IP address. To begin with, our approach would be to rely on the level of indirection provided by the fabric to optionally hide the sensor's network identity from subscribers.

2.2 Discovery and Publication

The naming and discovery service, which in many ways can be treated as a publish and subscribe mechanism, plays a similar role to that of the DNS in today's Internet. The service would map a tuple space of attributes to a handle (or set of handles) that may be used to operate on data streams. Attributes will typically consist of information such as the sensing modality, data format, location, orientation, etc.

There are two constraints that must be satisfied before returning handles to data streams. First, the attributes of the data stream and the attributes requested by the subscriber must match in some sense. There are some attributes that naturally form hierarchical relationships. Location is a primary example. These relationships must be known by the naming and discovery service. Second, disclosure rules accompanied by the data stream must be satisfied, which may take into account some aspect of the *subscribers* attributes (identity, location, and time).

Another challenge is to ease and encourage publication of sensor data by independent data providers, as well as application development by 3rd parties that pull on the published data streams. Primitive functionality should include aggregation, processing, and querying. The elements providing these services act as subscribers to data streams. In some cases, subscribers will act independently and crawl the network for available data streams, much like indexing services such as Google or Yahoo. In other cases, the sensor will task subscribers to aggregate data on its behalf, much like Flickr and blogging services.

3 The Partisans System Architecture

3.1 Architectural Components

Our proposed architecture incorporates the entities listed below.

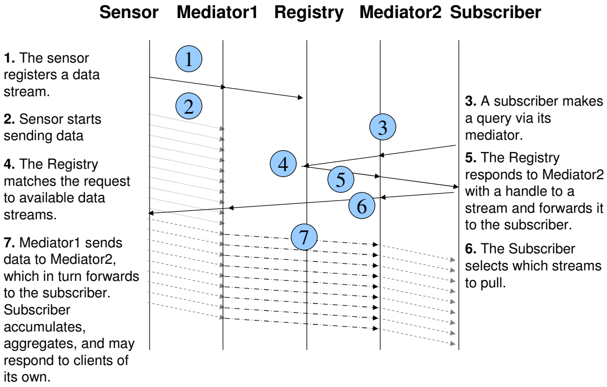
Sensors are sources of data at the edges of the network. They may either be resource rich devices (e.g. [10]) directly resident on the network as host nodes, or resource-constrained devices such as “Motes” [4], grouped many-to-one behind a resource rich proxy gateway node. We note that sensors can have roles beyond simply pure sources of data, providing control points at their interfaces to the external world for purposes such as configuring the sensor, providing global contextual information, and actuation in terms of local information output.

Subscribers are sinks of sensor data. They may be either individual users interested in data streams and event notifications from sensors, or network applications that subscribe to sensor data and provide archiving, aggregation, distillation, signal search, and other such services to their clients. A physician may subscribe to medical events associated with a remotely monitored patient; or a smart home control software may subscribe to data from sensors deployed by the homeowner. Network applications acting as hosting services could allow users to share sensor data much like Flickr.com (images), Vimeo.com (video) and Odeo.com (audio). Once hosting services exist, it is sensible to posit a Google-like service that facilitates searching archived sensor signals in terms of queries.

Mediators are nodes in the network that provide (under application control) selected in-network functions on sensor data streams. These functions would include: enhancing streams with attested contextual information at a specified resolution; performing verifications on sensor data values such as range checks or comparisons with values at proximate sensors; performing anonymization of the streams by removing device identification information; replicating streams for delivery to different nodes; and providing reliability for intermittently connected sensor and/or subscribers. The functions themselves are performed based on disclosure and verification rules specified by the sensor. Moreover, the mediator would make use of trusted infrastructure for independently measuring the location and time of data sent by the sensor devices.

Distinct from efforts such as Active Networks, the mediators do not manipulate the sensor data values carried as stream payload content, a choice motivated by the simplicity of not allowing complex applications to “program” the mediators. Transformation of sensor value streams for purposes such as anonymity preservation or for scaling to a presentation device is best delegated to the end points. We view the mediators to be like firewalls in terms of administration, deployment ubiquity, trust and transparency to the user, while being like distributed content caching servers in terms of inter-mediator coordination and hardware configuration.

Registries are network entities that help subscribers discover and bind with sensor data streams. Their role is to provide a service analogous to that of the Domain Name Service (DNS), with a model of administration and deployment-ubiquity similar to DNS servers. Sensors register with the registry metadata information about the sensor data they publish, while subscribers use the registry to search for sensor data streams based on queries over attributes such as location or type of sensor data. The registry maps the query via a tuple space search process to return a handle, or set of handles, to sensor data streams.



1. The sensor registers a data stream.

2. Sensor starts sending data

4. The Registry matches the request to available data streams.

7. Mediator1 sends data to Mediator2, which in turn forwards to the subscriber. Subscriber accumulates, aggregates, and may respond to clients of its own.

3. A subscriber makes a query via its mediator.

5. The Registry responds to Mediator2 with a handle to a stream and forwards it to the subscriber.

6. The Subscriber selects which streams to pull.

Figure 2: A sequence diagram of an example exchange between a sensor and subscriber.

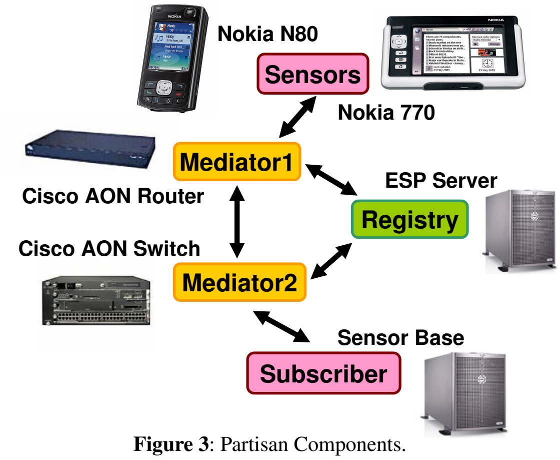


Figure 3: Partisan Components.

3.2 An Example Exchange

Figure 2 depicts an example exchange between a sensor and subscriber. The steps below go into more detail.

Step 1. Upon deployment and configuration by its owner, a sensor registers the streams it is publishing with the registry service via Mediator1. The registration contains information about the sensor type, location, and context. It is these attributes that others will use to search and subscribe to sensor data streams. The registration will also contain disclosure and verification rules. For example, a sensor device could register (location= “Main Library”, type= “microphone”, format= “spectrogram”, UID= “3241531”). The sensor contributes to the location information in the data stream attribute, since the location may be at a finer granularity than to which the fabric is able to attest. If the sensor is mobile, it is the sensor’s responsibility to detect when it may have changed locations and should bind with a new mediator and update the registry.

Step 2. The sensor now begins to publish its data to Mediator1, either proactively or when demanded by Mediator1 (as a consequence of a request by a subscriber).

Step 3. Sometime later the Subscriber sends a query to the Registry through its mediator, Mediator2. The request contains disclosure rules as well. It needs to go through Mediator2 in order for the fabric to attest to the location (or similar attributes) of the requestor, as the sensor may have disclosure controls that are a function of requestor’s attributes.

Step 4. The Registry reconciles the disclosure rules of the sensor publishing the data with the request and attributes of the subscriber.

Step 5. The Registry returns a pointer to the matching sensor data streams to Mediator2. These pointers direct the holder to mediators that proxy the corresponding data streams.

Step 6. The Subscriber then directs Mediator2 as to which particular data streams to pull. (In the figure, there’s only one stream to pull.)

Step 7. Mediator2 then pulls the requested data streams from the originating mediators.

3.3 Prototypes

We have prototype implementations of our four system components, as depicted in Figure 3.

The prototype sensor platforms are the Nokia N80 smart phone, and the Nokia 770 Internet Tablet. An example application that uses these platforms is the EcoPDA (Ecological PDA) Project [2], an effort underway to assist in field observations for biodiversity and ecological research. With little modification, however, the EcoPDA project can satisfy the role of the Sensor in PSUS applications. The goal of EcoPDA is to automate or augment many aspects of mobile data entry (GPS location, voice recording, imaging) as well as prompt for and verify data input by the field researcher. Additionally, the EcoPDA will upload to an aggregator, such as SensorBase[3] (described later in this section) on behalf of the user.

The prototype mediator rests upon the AON (Application Oriented Networking) platform. In [1], Sankar describes how AON platforms contribute to the architecture of intelligent edge networks. The salient features of AON routers and switches are that they do application-level message classification, provide a declarative policy framework within which to operate on these messages (drop, cache, modify, forward), and an interface to easily inject policy into the network. We are currently developing software running on an AON Volant Blade to do simple location testimony/verification and data aggregation/obfuscation.

The prototype registry is built upon the ESP framework. [12] ESP brings forward several concepts that are essential to managing, querying, and interacting with the wide variety of network sensing systems. The unifying interface language is ESPml. Sensor systems register themselves by describing their capabilities as an ESPml document. Agents can query the registry based on an area of interest and are returned an ESPml document that contains all the systems that exist in the query space.

The prototype aggregator is SensorBase.org [3], which is a platform for common data storage and management for sensor networks. It provides users a web-service interface for publishing sensor network data. SensorBase also acts as a sensor network specific search engine, which allows users to query for specific data sets based on geographic location, sensor type, date/time range, and other relevant fields. Furthermore, the ability to search based on characteristics or features of the data themselves will soon be added.

4 Conclusion

We have presented an architecture for infrastructure supported selective data sharing and verification. Network testimony of when and where data is first injected allows mediating infrastructure nodes to execute selective data sharing on behalf of data contributers, and verification functions on behalf of data consumers. The result is an audit trail that plausibily verifies the integrity the sensor data and some degree of context around that data (time and location). These services are a requirement for a rapidly emerging class of applications that draw upon sensed information about people, objects, and physical spaces.

There is a natural resistence to introducing yet more functionality into the network infrastructure, however, the services we propose (time and location testimony) are aspects of communication to which the network already has access; our proposal is to expose and utilize this information in novel ways. Though theoretically it's possible to achieve the same effects of verification and selective sharing end-to-end (possibly with crytographic techniques like zero knowledge proofs), our architecture represents a trade-off between trust and practicality, taking into consideration the often limited resources of the sensor.

References

[1] Cisco application-oriented networking blog. `http://blogs.cisco.com/AON/`.

[2] Ecopda. `http://www.lecs.cs.ucla.edu/urban-sensing/index.php/EcoPDA`.

[3] Sensorbase. `http://research.cens.ucla.edu/projects/2006/Systems_Infrastructure/SensorBase/default.htm`.

[4] Tinyos: An operating system for networked sensors. *http://tinyos.millennium.berkeley.edu*.

[5] Ucla cs219: Urban sensing applications. `http://http://www.lecs.cs.ucla.edu/urban-sensing/index.php/CS219_Spring_2006`.

[6] Ganeriwal, S., Kumar, R., and Srivastava, M. Timing Sync Protocol for Sensor Networks. In *Sensys* (Los Angeles, 2003).

[7] Huffman, S. M., and Reifer, M. H. Method for geolocating logical network address. In *United States Patent 6947978* (Sept. 2005).

[8] Kansal, A., and Srivastava, M. *Wireless Sensor Networks: A Systems Perspective, Eds. N. Bulusu and S. Jha*. Artech House, 2005, ch. Energy harvesting aware power management.

[9] Kohno, T., Broido, A., and Claffy, K. C. Remote physical device fingerprinting. *IEEE Trans. Dependable Secur. Comput. 2*, 2 (2005), 93–108.

[10] McIntire, D., Ho, K., Yip, B., Singh, A., Wu, W., and Kaiser, W. J. The low power energy aware processing (leap)embedded networked sensor system. In *IPSN 2006* (New York, NY, USA, 2006), ACM Press, pp. 449–457.

[11] Mills, D. L. Internet Time Synchronization: The Network Time Protocol. In *Global States and Time in Distributed Systems*, Z. Yang and T. A. Marsland, Eds. IEEE Computer Society Press, 1994.

[12] Reddy, S., Schmid, T., Parker, A., Porway, J., Chen, G., Joki, A., Burke, J., Hansen, M., Estrin, D., and Srivastava, M. Demo: Urbancens: Sensing with the urban context in mind. In *Ubicomp, to appear* (2006).

[13] Savvides, A., Girod, L., Srivastava, M., and Estrin, D. Localization in sensor networks. In *Wireless Sensor Networks*, C. S. Raghavendra, K. M. Sivalingam, and T. Znati, Eds. Kluwer, 2004.