Sensor Cloud – Sensing As a Service Paradigm

Sanjay K Madria
Curators' Distinguished Professor
ACM Distinguished Scientist
Director, Web and Wireless Computing Lab
Department of Computer Science
madrias@mst.edu

WEB & WIRELESS COMPUTING LABORATORY
MISSOURI UNIVERSITY OF SCIENCE AND TECHNOLOGY

Funded by NIST with papers in IEEE SOFTWARE, 2014 (100+ citations), IEEE MDM 2014, IEEE SRDS 2014, IEEE SRDS 2015 (Best paper award), DASFAA 2015, IEEE TSC, 2016, etc.. Book by Morgan-Claypool in Sept. 2018.

Introduction

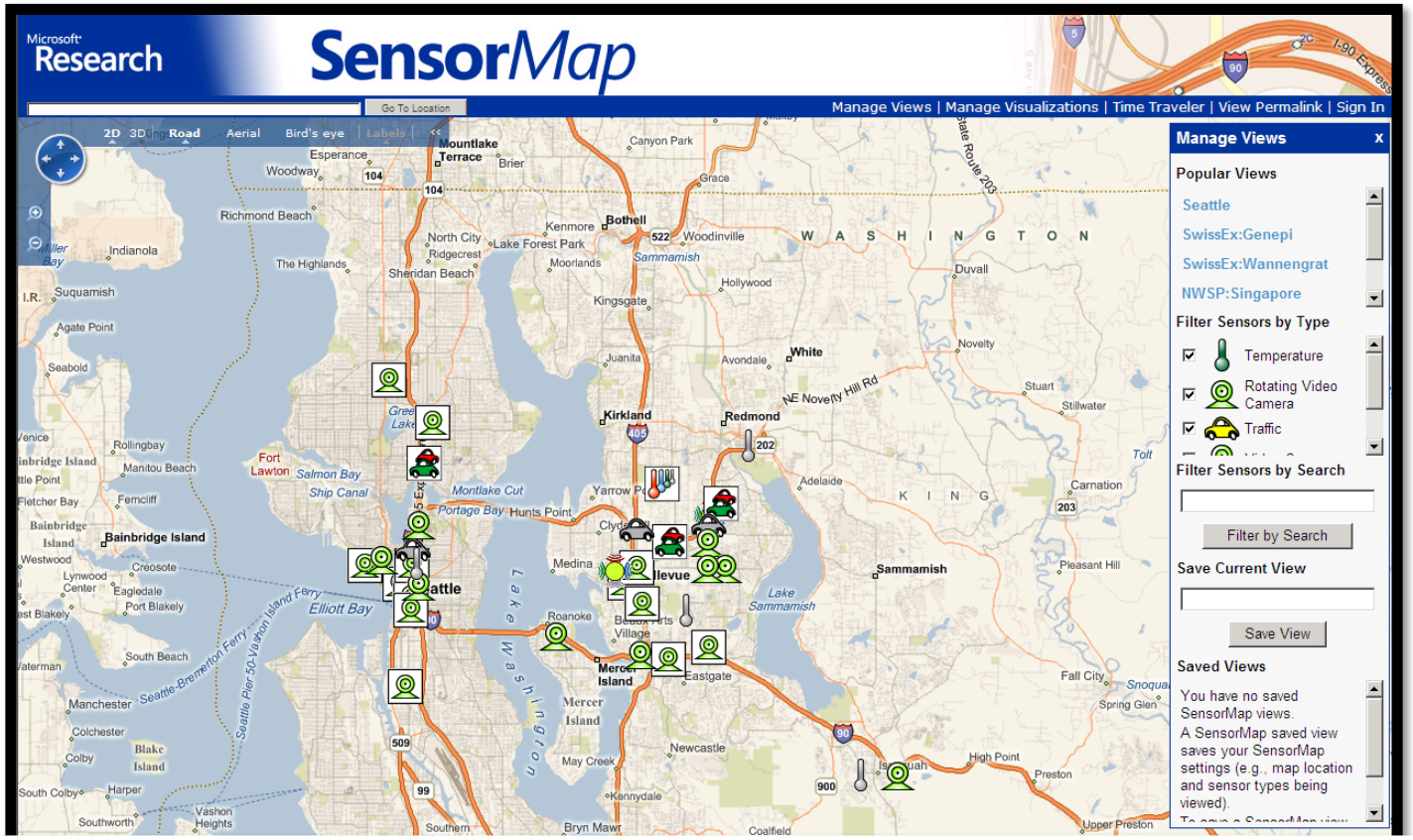
- Wireless Sensor Networks (WSNs) have gained popularity for applications like disaster management, health and environmental monitoring, IoTs, CPS, etc.)
  - Small size, Ease of deployment and Self-organization
- Challenges to medium to small scale industries
  - cost of owning and maintaining WSNs
  - under utilized WSNs affect profitability

- *Sensor Cloud Computing* platform has been proposed: a model for enabling ubiquitous, on-demand access to a shared pool of configurable sensors that can be rapidly provisioned and released with minimal management effort or WSN provider interaction

More on Sensor Cloud Applications

- Sensors are continually recording information about weather, land use, vegetation, precipitation, drought, water quality, etc. which can support multiple applications on demand
- Tracking correlations between datasets both based on geospatial and temporally for biodiversity changes, invasive species, and at-risk species etc. – need collaborative environment and scalability
- Can collect data via scheduling and many different users can use sensing services at the same time
- Multi-task Learning - play with dimensions, different features, sampling rate as different WSNs have different policies

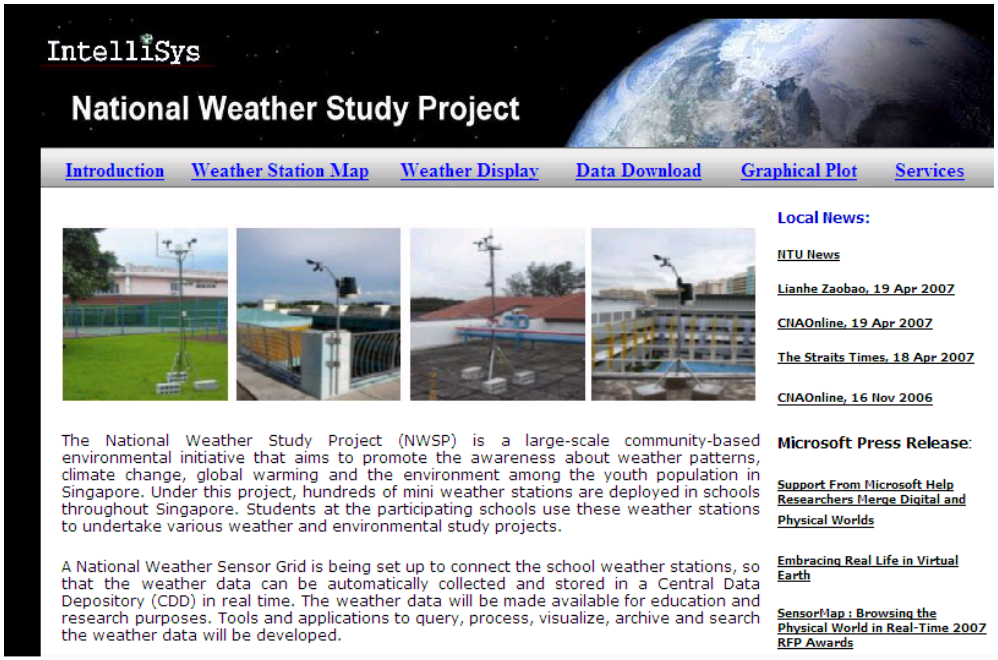
Motivation: Players in this game



SensorMap targets a new class of applications that relies on real-time sensor data and its mash-up with the geocentric Web to provide instantaneous environmental visibility and timely decision support.

Motivation: Players in this game

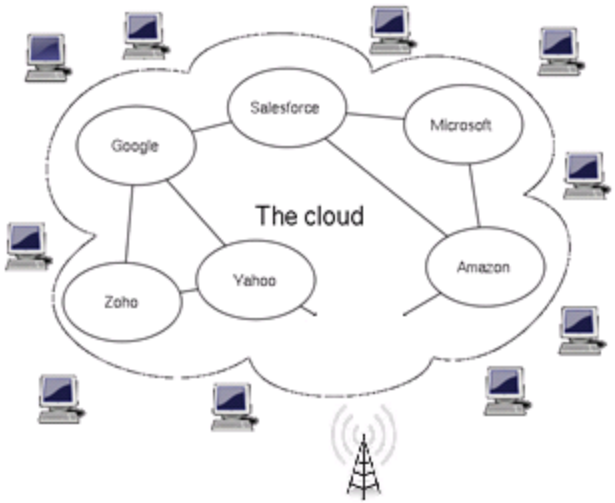
- Asia-Pacific Environmental Sensor Grid (APESG)



APESG was formed with an attempt to make use of data sensed by sensors or other monitoring systems in the participating APEC economies. The initiative also provides a conduit between field practices, scientists, and policy makers to share and/or exchange expertise and experience. This, in turn, may be extended as an early warning response to mitigate potential environmental disasters.

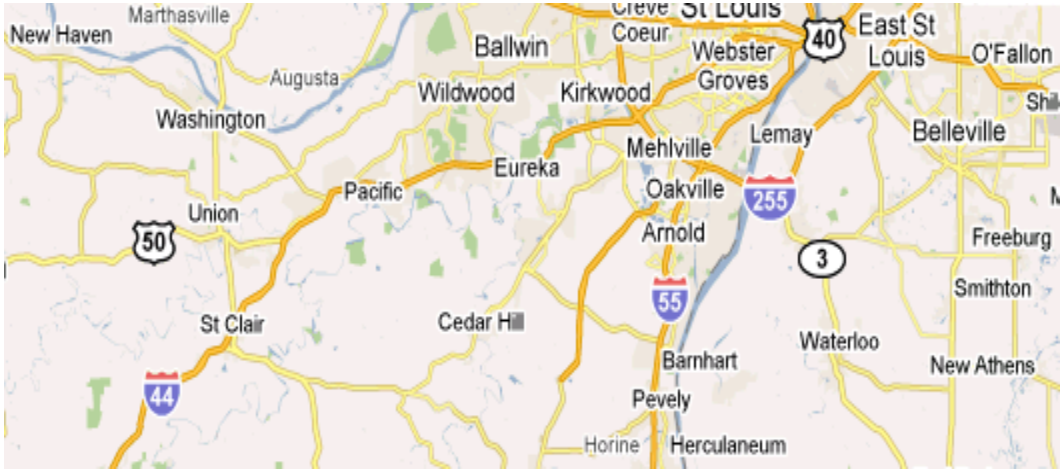
Motivation - Sensor Cloud Concept

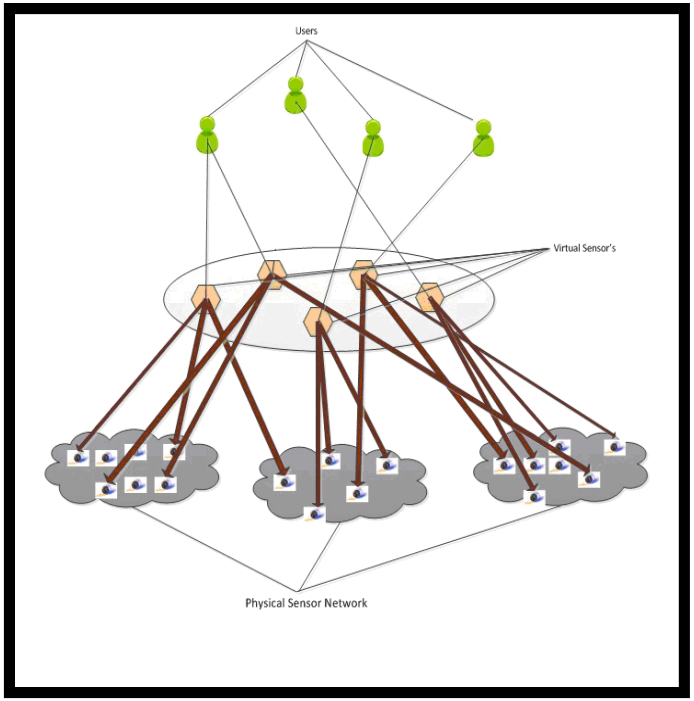
- Research in several disciplines have shown significant interest in the development of virtualization and cloud computing
- This made possible sensor-cloud computing:



What is Sensor-cloud Computing

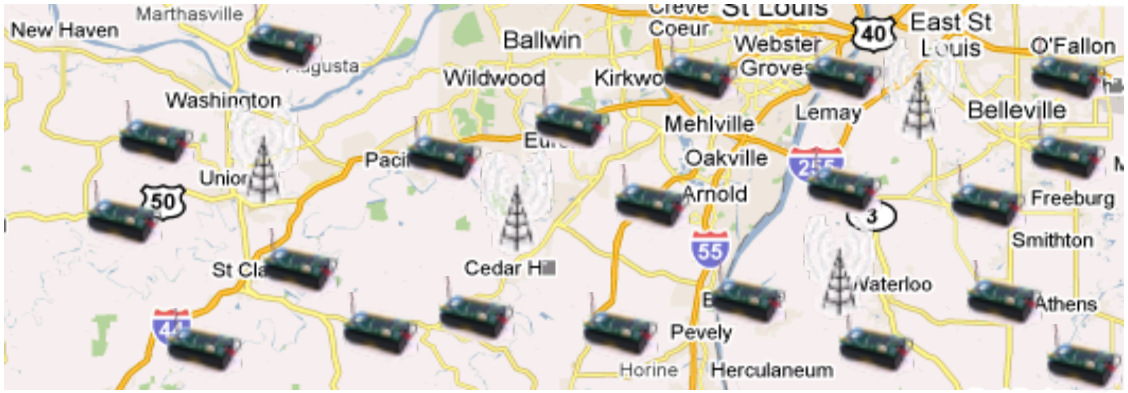
- A heterogeneous computing environment in which there are potentially millions of deployed sensors
- A group of sensors is operated by individual organizations



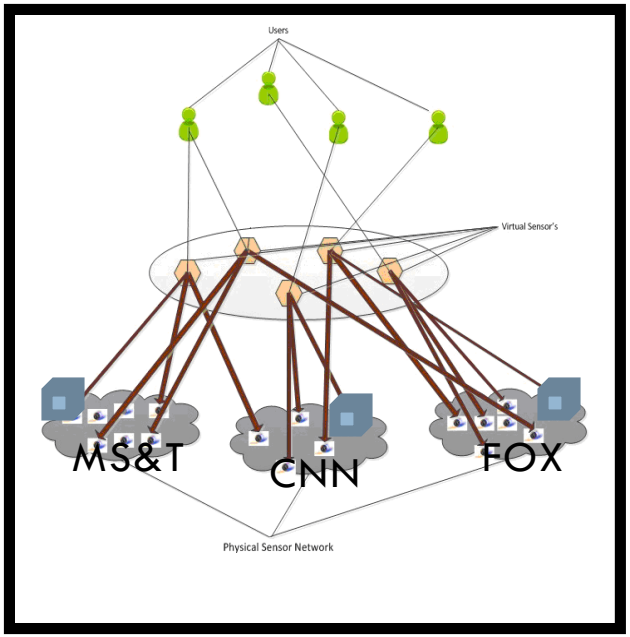


What is Sensor-cloud Computing?

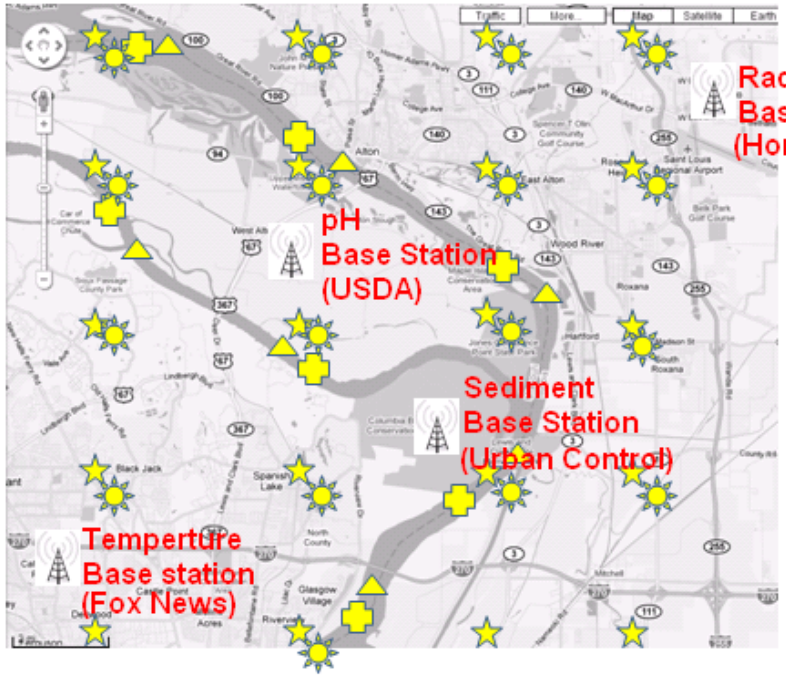
- The sensors may continue sensing a steady stream of information or passively report the observation
- Data are relayed through a gateway and middleware layer network, where sensing data is consumed by the **"sensor-cloud infrastructure"**

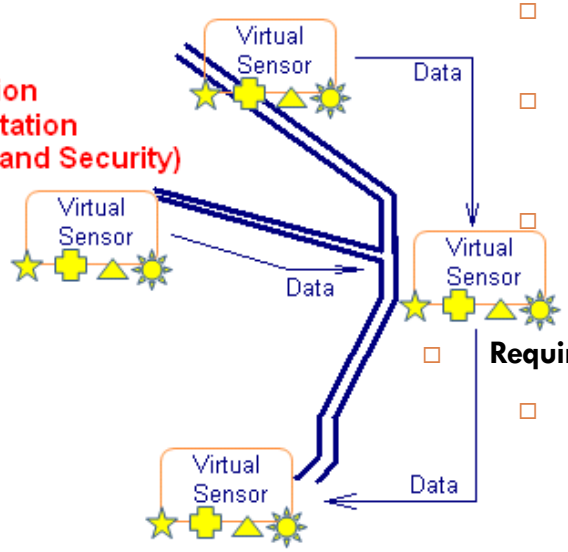


Madria et al., IEEE Software 2014



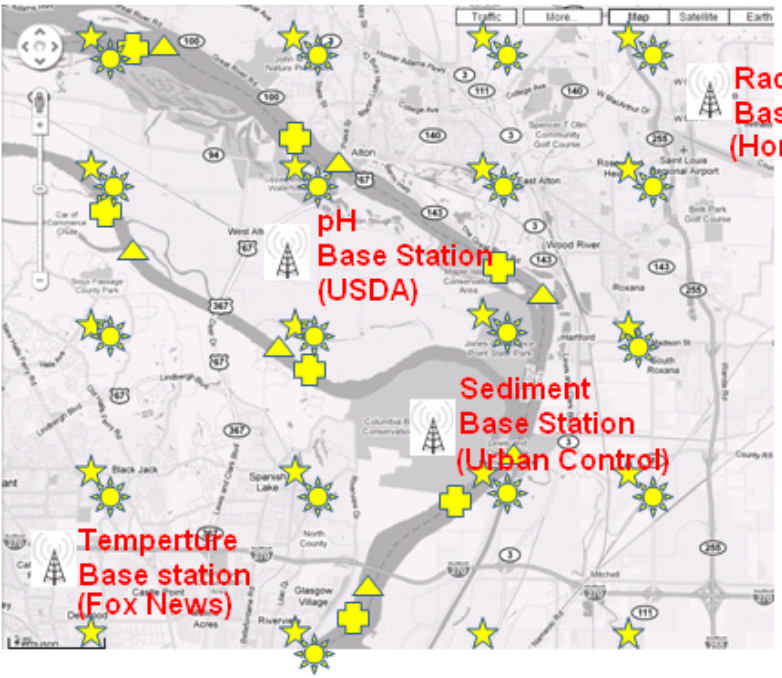
Motivating Scenarios: Chemical Tracking

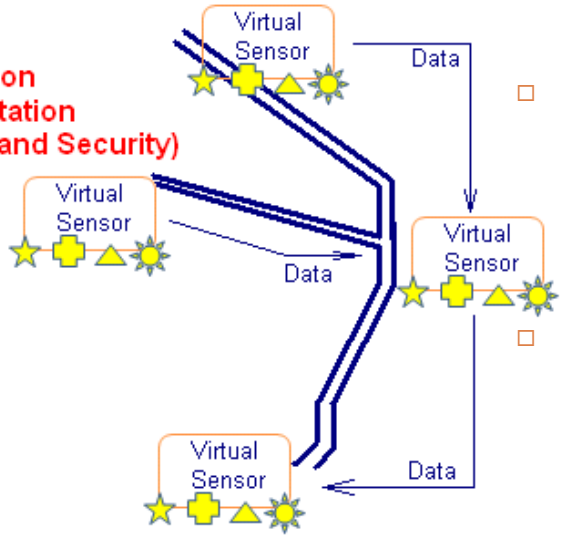




- **Scenario**
  - Chemical X spill from the upstream factory
  - It is hazardous to and no known sensors capable to detect 'X' deployed along the river
  - Specific sensors can be produced but we need to form a sensor network **immediately**
- **Requirements**
  - Need a **multi-sensing** sensors that capable of detecting unique characteristics to detect chemical X
  - Need to form a sensor network in such a way to help researcher assess the damage using a **Network flow model**
  - These sensors need to collaborate in such a way that the down stream sensors **require inputs from** upstream sensors in its statistical assessment and analysis

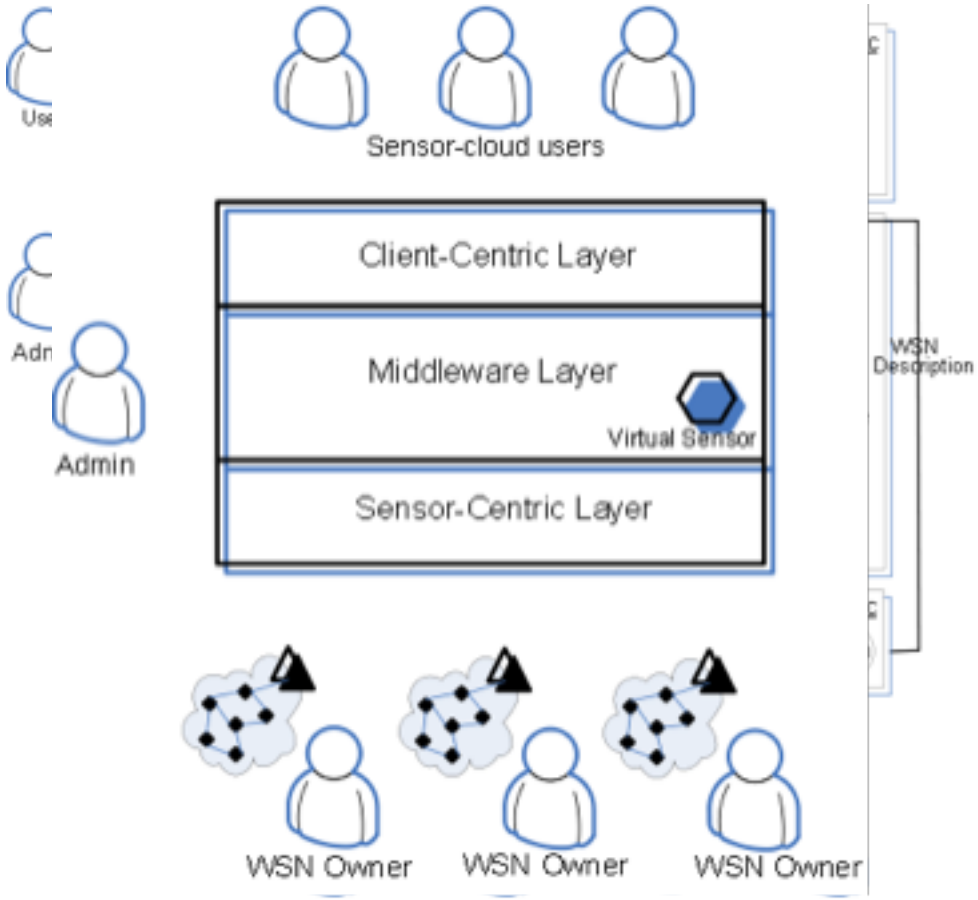
Motivating Scenarios: Chemical Tracking





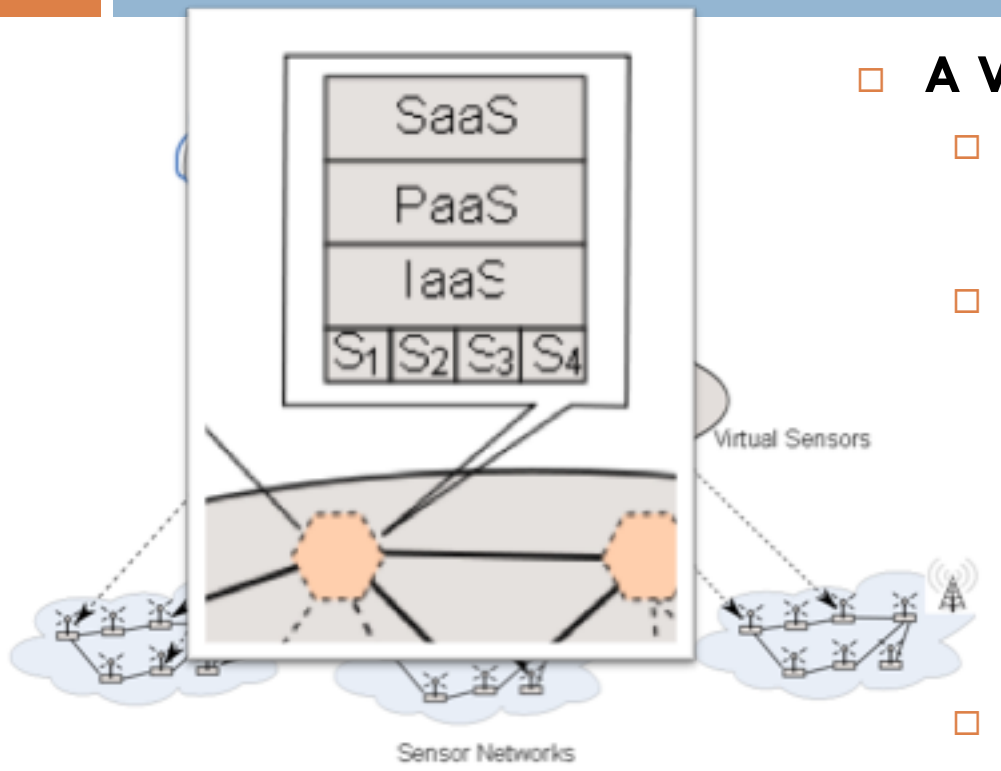
- **Advantage of Sensor-cloud**
  - Significantly reduce TCO
    - Simply rent the sensor
  - Resource Sharing
    - Scale up or down as needed
    - No need to engineer the peak load
  - Platform independent
    - No need to know how to implement micaz or TelosB
    - You only know what do you want to sense → Ease of use
  - Greater computation and data collaboration with the server computation capability at the virtual level
  - Location freedom
    - You can access them from every using internet

Sensor-cloud Architecture



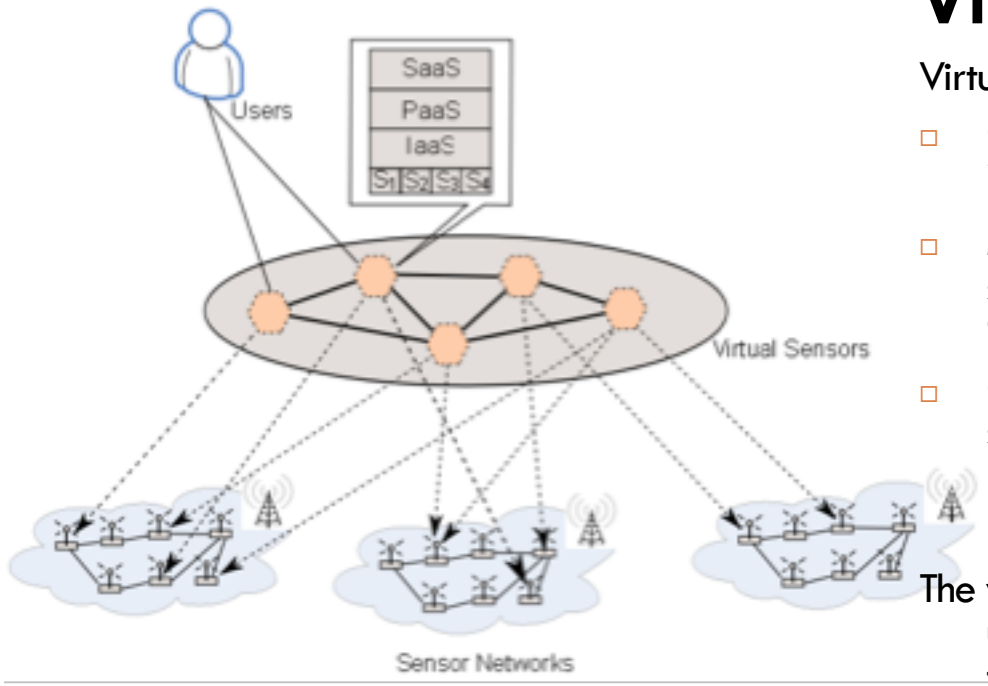
- **Client-Centric Layer**
  - A service gateway to the working session and mission management
  - A service gateway to the virtual sensors
- **Middleware Layer**
  - Administrates:
    - Service Negotiation
    - Virtual sensor fabrication
    - Virtual network deployment
    - Service life-cycle
    - Accounting (e.g., Billing etc.)
- **Sensor-Centric Layer**
  - Provides the DNS-like function to provision the inter connectivity between sensor-cloud and the physical sensor network

Virtual Sensors



- **A Virtual Sensor**
  - An image of an application with the sensing capabilities inherited from physical sensors
  - A virtual sensors act as a Software as a Service (SaaS) which is responsible for processing the data which is sensed by the physical sensor
    - User is projected with an illusion that the instance given to him is the physical sensor which does the sensing
  - Platform as a Service (PaaS) provides the computation, and repository environment for SaaS
  - Infrastructure as a Service (IaaS) provides the gateway between virtual and physical sensor

Virtualization



Virtualization

Virtualization can be done in many ways.

- **One-on-one virtualization:** when one physical sensor is mapped to one virtual sensor
- **Many-to-one virtualization:** when multiple physical sensors are mapped to one virtual instance. This enhances a virtual sensor with multi-sensing capability
- **One-to-many virtualization:** multiple users share the same physical sensors, but through their own instance allocated to them by the middleware

The virtualization projects to the user that they are the sole user of that corresponding physical sensor as well as the abstraction that the user is interacting with the physical sensors directly

The commands issued by the user are passed on to the physical sensors by the middleware. The group which the user creates is visible to him and the administrator who handles the application

Service Lifecycle

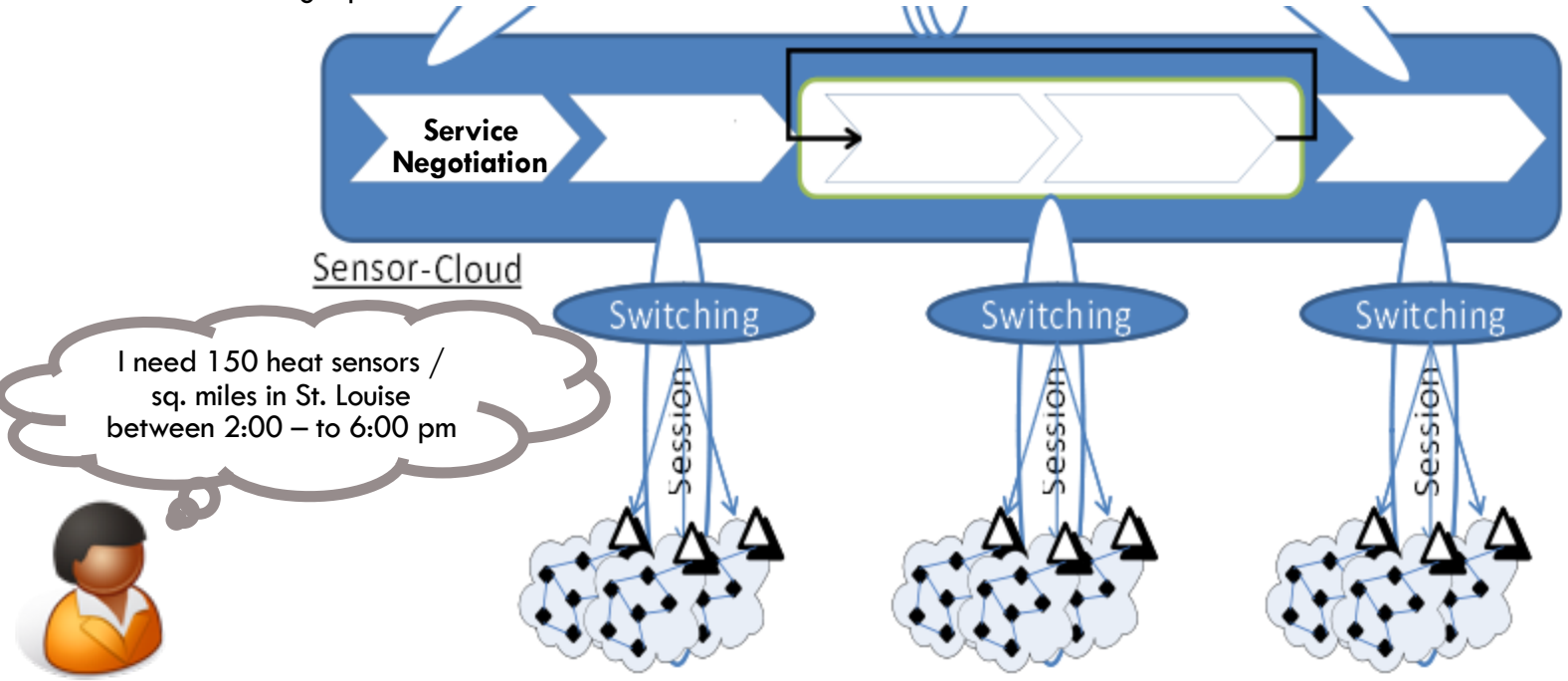
Service Lifecycle

In sensor-cloud computing, users do not need to know what types of physical sensors are deployed.

- They only need to know what they want to **use the them for**

Service Negotiation

- The process between users and the sensor-cloud where the user challenges the sensor-cloud service with the use scenario
- The use scenario is described as a set of expectations on the environment in which the user want to do the sensing operation



Service Lifecycle

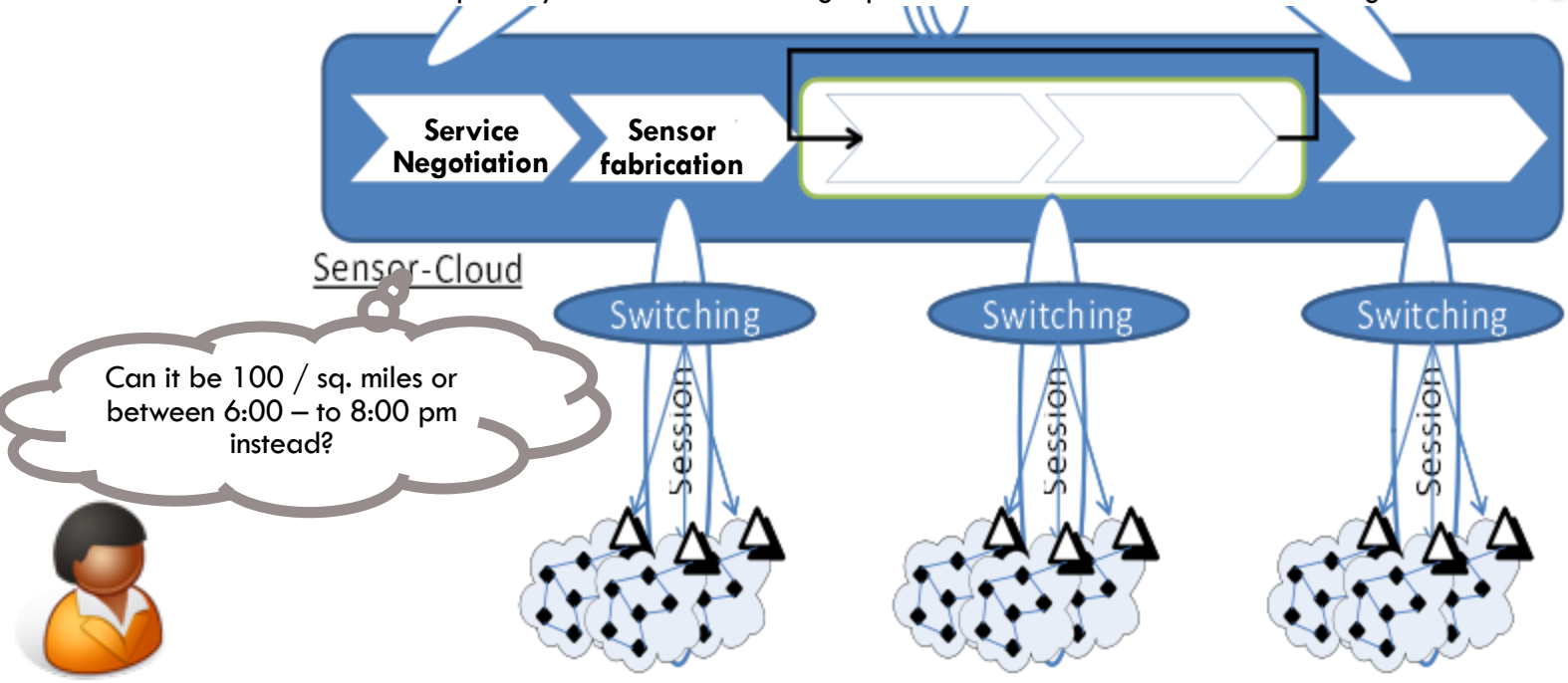
Service Lifecycle

In sensor-cloud computing, users do not need to know what types of physical sensors are deployed.

- They only need to know what they want to **use the them for**

Service Negotiation

- Assuming an existence of the service discovery unit which is capable of locating the sensors that meet the criteria, sensor cloud can determine if it could host such a service → Service Negotiation
- Upon agreement, the sensor-cloud instantiates virtual sensors and delegate the ownership to a user so as to allow him to temporary control the sensing operation via the transaction management → Fabrication



Service Lifecycle

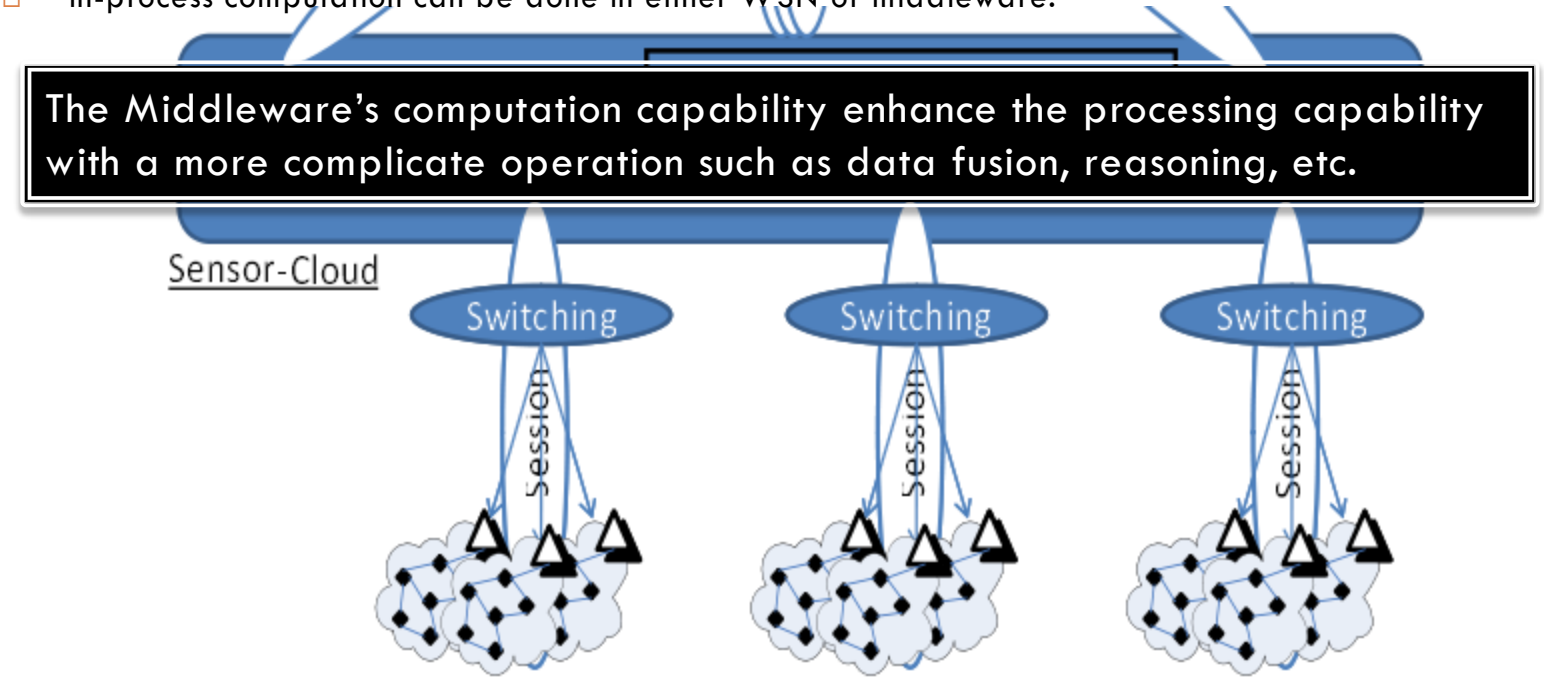
Service Lifecycle

In sensor-cloud computing, users do not need to know what types of physical sensors are deployed.

- They only need to know what they want to **use the them for**

Sensing Mission

- Sensing data is collected by the physical sensors where the sensor-cloud ensure freshness of sensing operation through the sensor network audition and standard code of conduct
- The readings are collected by virtual sensors where in-process operation can be executed
- In-process computation can be done in either WSN or middleware.



Service Lifecycle

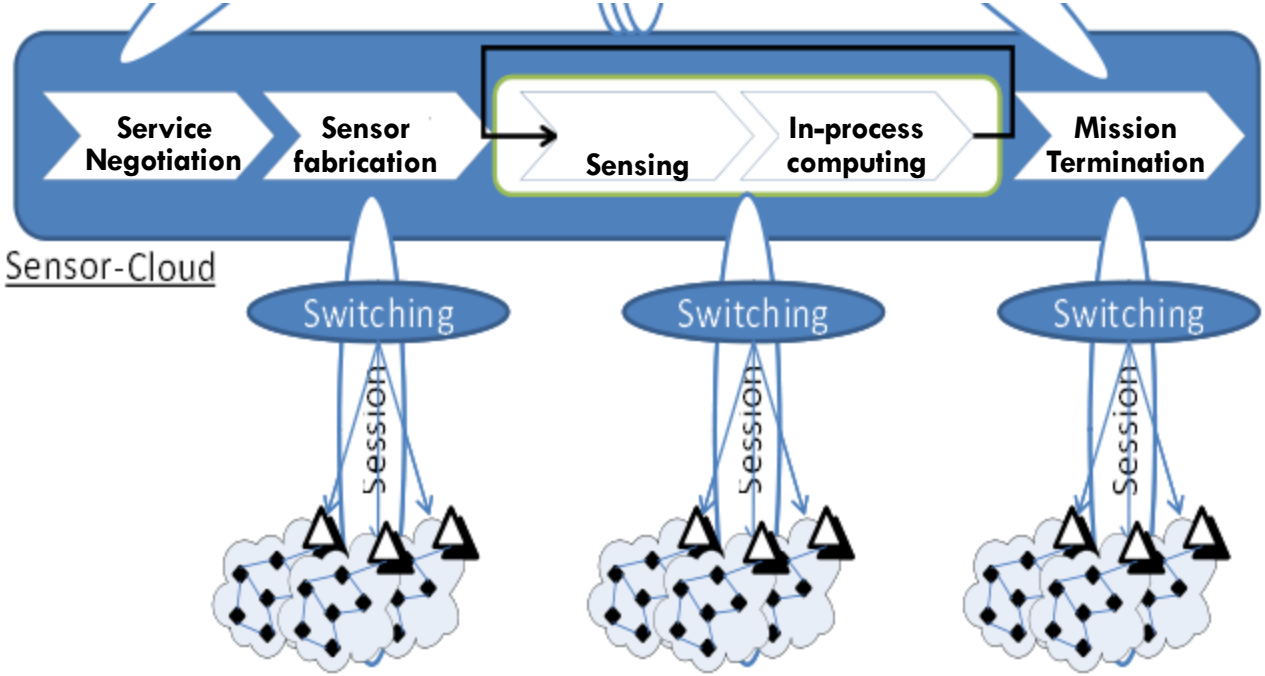
Service Lifecycle

In sensor-cloud computing, users do not need to know what types of physical sensors are deployed.

- They only need to know what they want to **use the them for**

Mission Termination

- When the mission concludes, the final result is ported to the end user and the whole session is terminated
- The sensor-cloud place the invoice and the physical sensors are released from the mission
- Sensor-cloud users do not have deal with the maintenance issues to the physical sensors



Scheduling Problem in Sensor Cloud

- Multiple WSN owners collaborate together to provide a sensor cloud service.
- The environment provides flexibility and energy savings to users to receive sensor data at different sampling time intervals by sharing
- Users may like to consume data in following ways:
  - Data at specific frequency
  - Ad-hoc data requests
  - Event based data
- a challenge to schedule WSNs such that it serves maximum number of users for different types of tasks

Rashmi Dalvi and Sanjay Madria, Energy Efficient Scheduling of Fine-granularity Tasks in a Sensor Cloud, in proceedings of 20th International Conference on Database Systems for Advanced Applications, Hanoi, Vietnam, (DASFAA) April, 2015

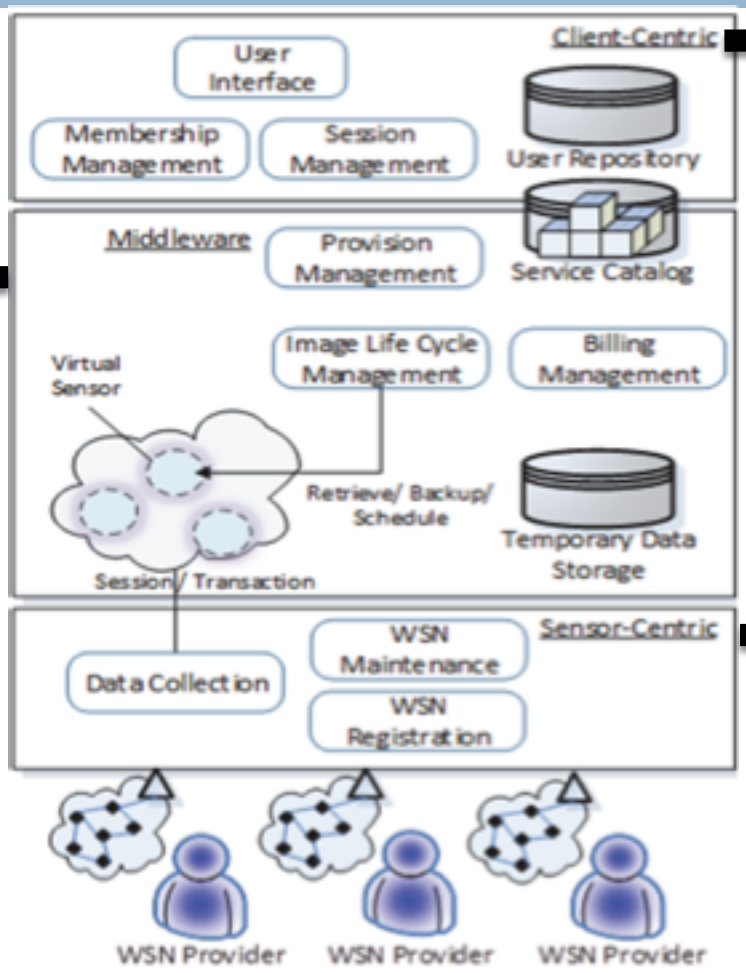
Types of Tasks

- **Type 1 – T1: PUSH Requests (data at a specific frequency)**
  - **Most expensive when used at short time interval**
  - **Preempts other requests**
  - **Inputs – location, sensing phenomena, duration, frequency**
  - **For specific set of sensors the operating frequency is minimum frequency among all requested frequencies of task T1**
- **Type 2 – T2: PULL Requests (one time data)**
  - **Ad-hoc requests**
  - **Inputs – location, sensing phenomena**
- **Type 3 – T3: Event Based requests**
  - **Inputs – location, sensing phenomena, condition to monitor, duration, frequency**
  - **Based on notifications this type is further classified as**
    - **Notify once (less expensive)**
    - **Notify until condition is false(more expensive) –as expensive as T1 on event occurrence**

Allocation Problem in Sensor Cloud

- Minimize the energy consumption to keep most of the sensors off where not needed.
- Cloud of heterogeneous sensor networks has sparsely and densely deployed WSNs working together.
- In dense zones, it is unnecessary to turn on all the sensors, because sensors in the vicinity tend to sense same data.
- In sparsely deployed zones, it is expected to allocate as many sensors as possible.
- Use spatial correlation, it is important to avail a solution to turn only the required sensors on.

Sensor Cloud Architecture



Client centric layer acts as a gateway between the sensor cloud and user. It consists of components facilitating and managing interactions between the user and core of the sensor cloud

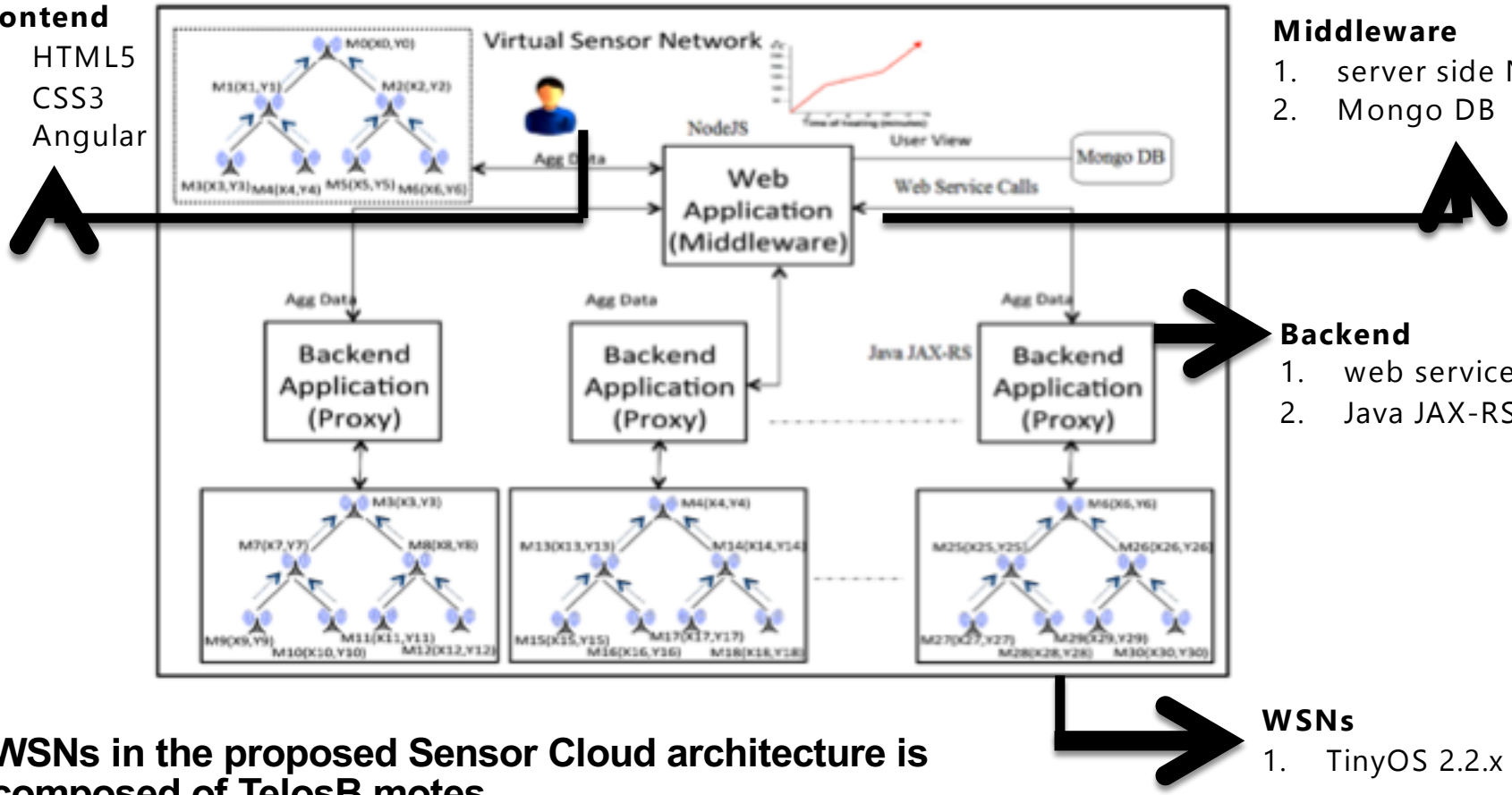
Middleware acts as an intermediary between client centric and sensor centric layers. It connects client requests with the data sent from the physical WSNs

Sensor centric layer assists in communicating requests coming from the Middleware to the physical WSNs.

Sanjay Madria, Amartya Sen, Rashmi Dalvi, A Sensor Cloud Test-bed for Multi-Model and Multi-User Sensor Applications, in the proceedings IEEE WCNC, 2016

System Setup & Implementation

Frontend
1. HTML5
2. CSS3
3. Angular



Middleware
1. server side NodeJS
2. Mongo DB

Backend
1. web services
2. Java JAX-RS

WSNs
1. TinyOS 2.2.x

WSNs in the proposed Sensor Cloud architecture is composed of TelosB motes

Sensing capabilities – humidity, temperature, light intensity, IR sensor, and vRef sensors

Multi-User Environment

- **Users request middleware for data from a specific WSN with required phenomenon and sampling time interval.**
- **Base station server validates the incoming request**
- **WSNs are moved to active mode and network undergoes topology discovery phase**
- **WSN are formed in accordance to a hierarchical tree structure**
- **Depending on a user's request, middleware displays data to each user according to their requested frequency, time duration, and sensing phenomenon**

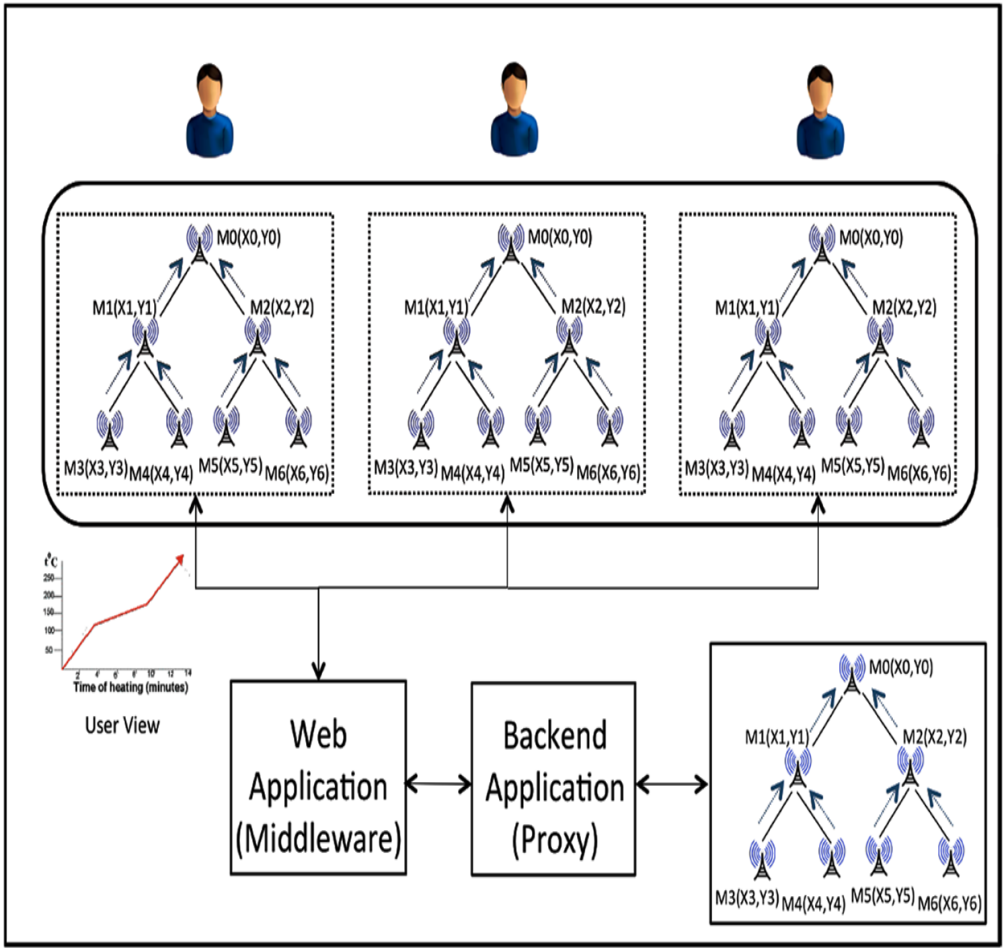


Fig: Multi user access to a WSN

Virtual Sensors

- **Internally aggregated data from WSN is represented as data for a specific virtual sensor**
- **Virtual sensors represents geographical region(s) selected by a user**
- **When user needs data from a region which is a collection of multiple WSNs, the middleware acknowledges the requests and forwards activation message to associated base station servers. Data from these WSNs are aggregated and displayed to the user as per their request**
- **This approach provides location transparency to users by hiding details about physical locations of WSNs, while dispensing their service requests.**

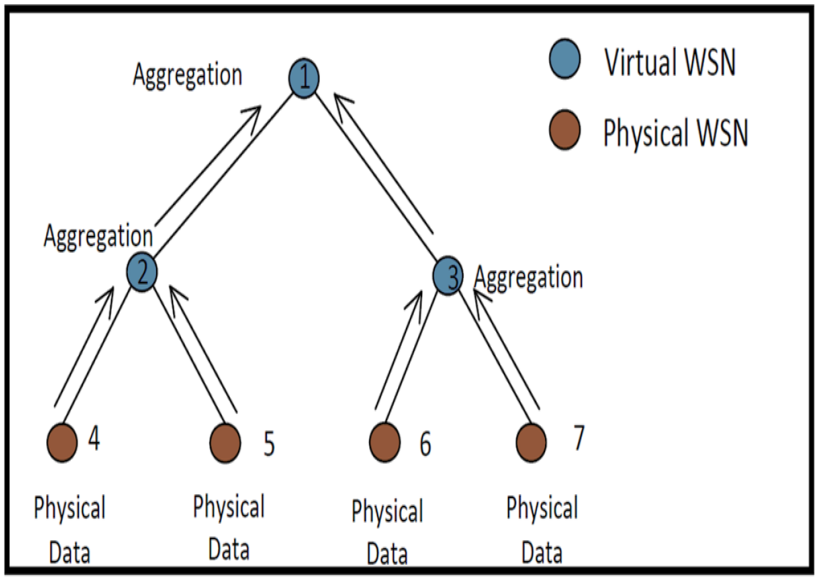


Fig: Hierarchy of network where each leaf level node represents a WSN. Other nodes represent regions. Order of data aggregation for virtual sensors depends on its position in the tree topology.

Sensor Cloud GUI

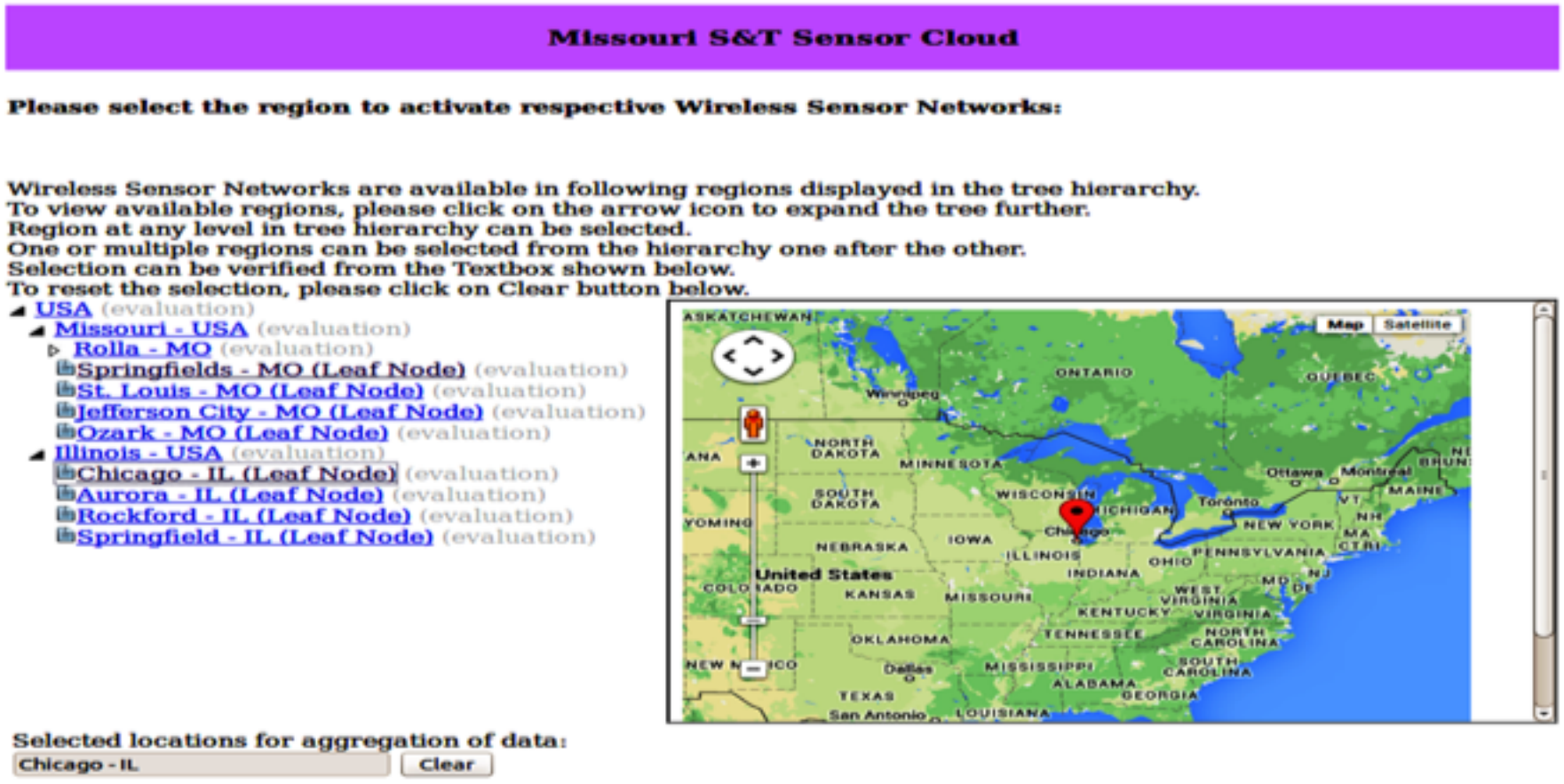


Fig: Sensor Cloud GUI for selection of Region of Interest. Tree structure on the left hand side of the GUI displays available regions having WSN services

Sensor Cloud GUI

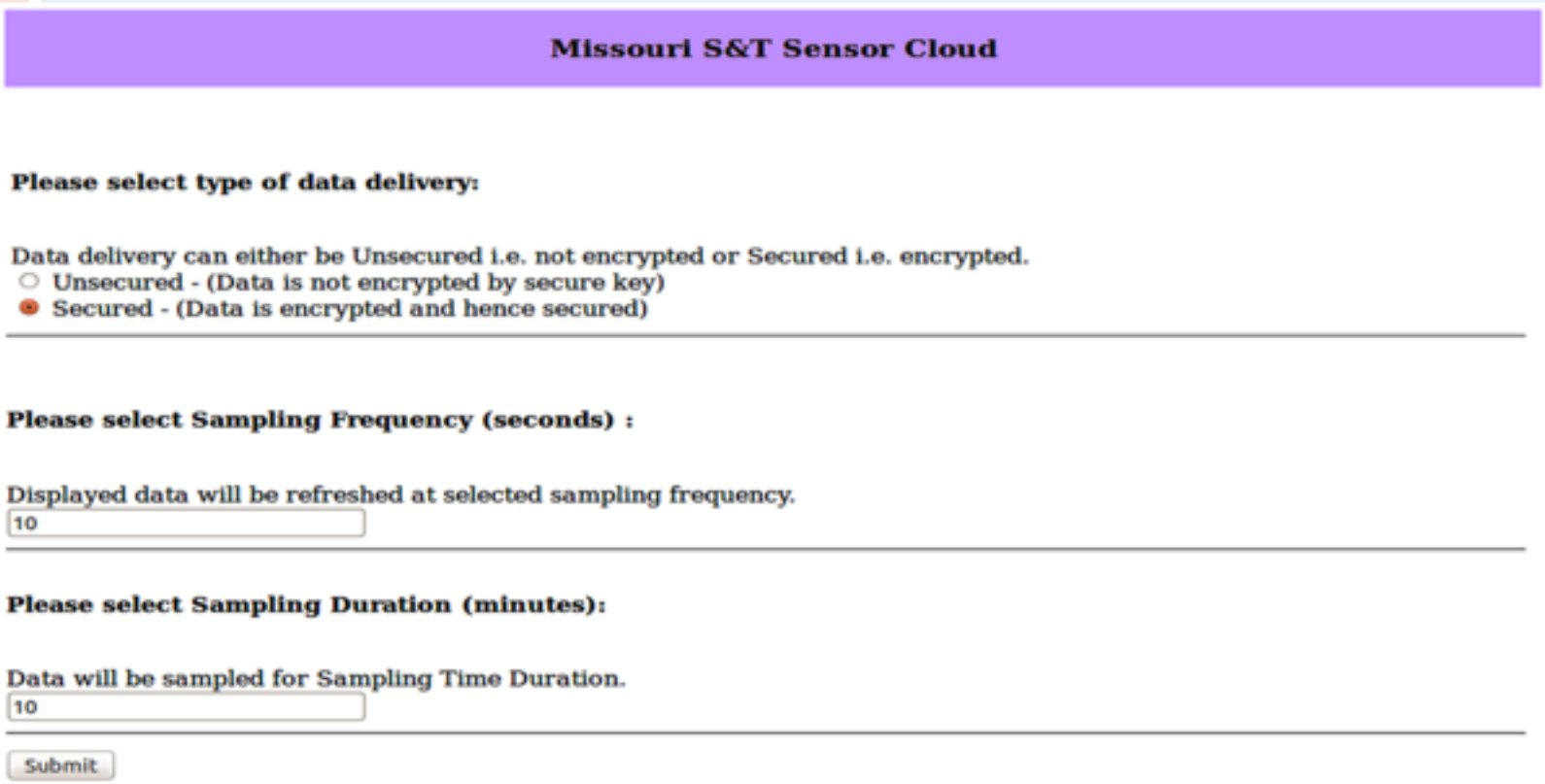


Fig: Sensor Cloud GUI for selection of service attributes. Users specify their desired security options, time frequency at which data will be sensed, and duration of service

Sensor Cloud GUI

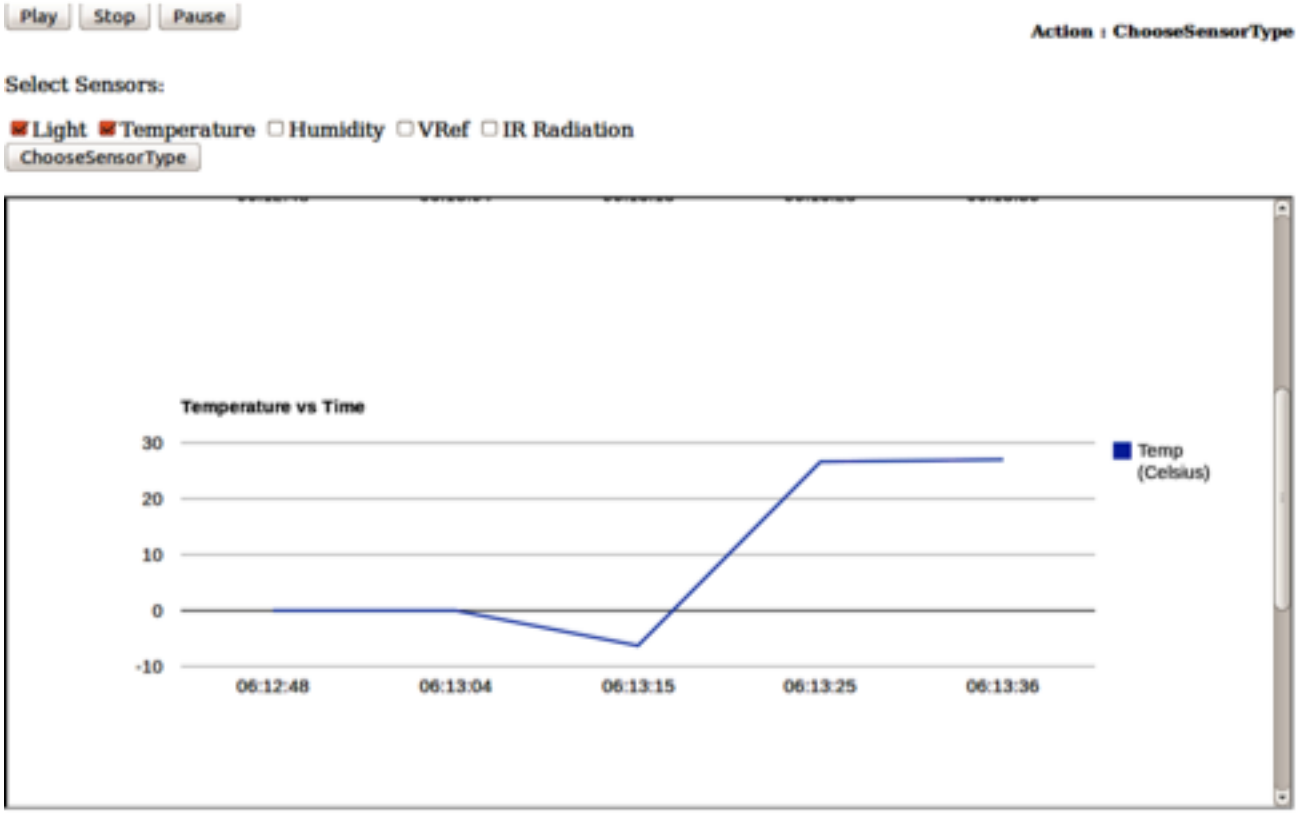
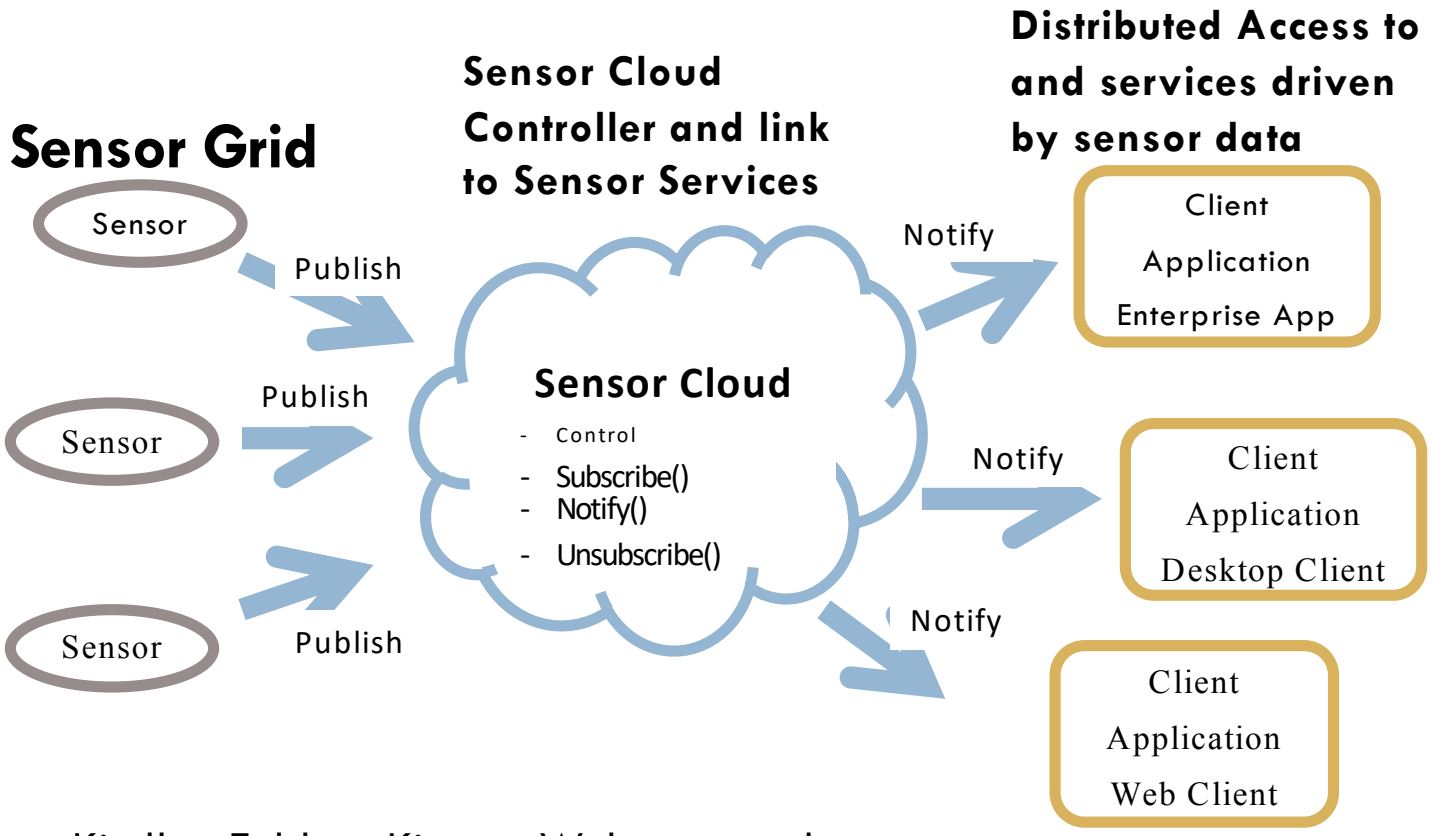


Fig: Sensor Cloud GUI displaying the received sensed data in real-time in graphical format

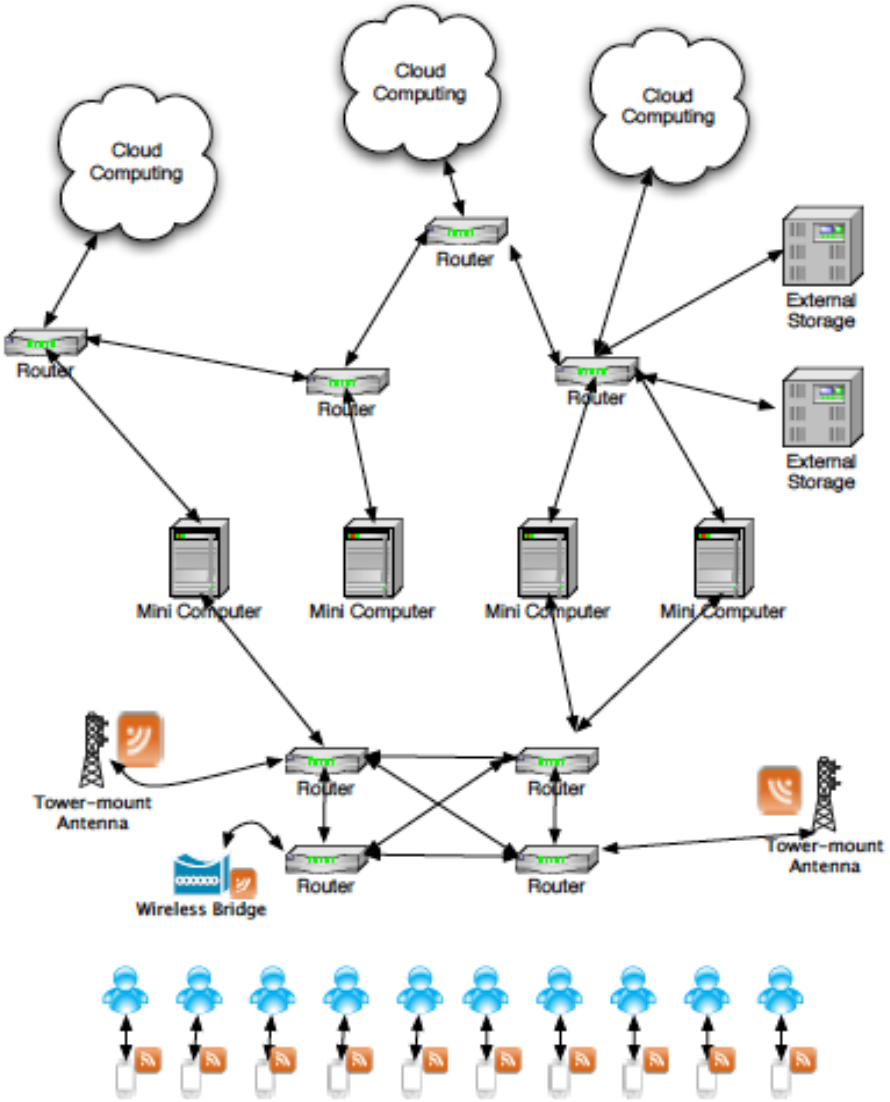
Internet of Things: Sensor Grids supported by Sensor Cloud



- Smart phones, Kindles, Tablets, Kinects, Web-cams, robots, are sensors
- Pub-Sub Brokers are cloud interface for sensors
- Filters subscribe to data from Sensors
- Naturally Collaborative
- Rebuilding software from scratch as Open Source – collaboration welcome

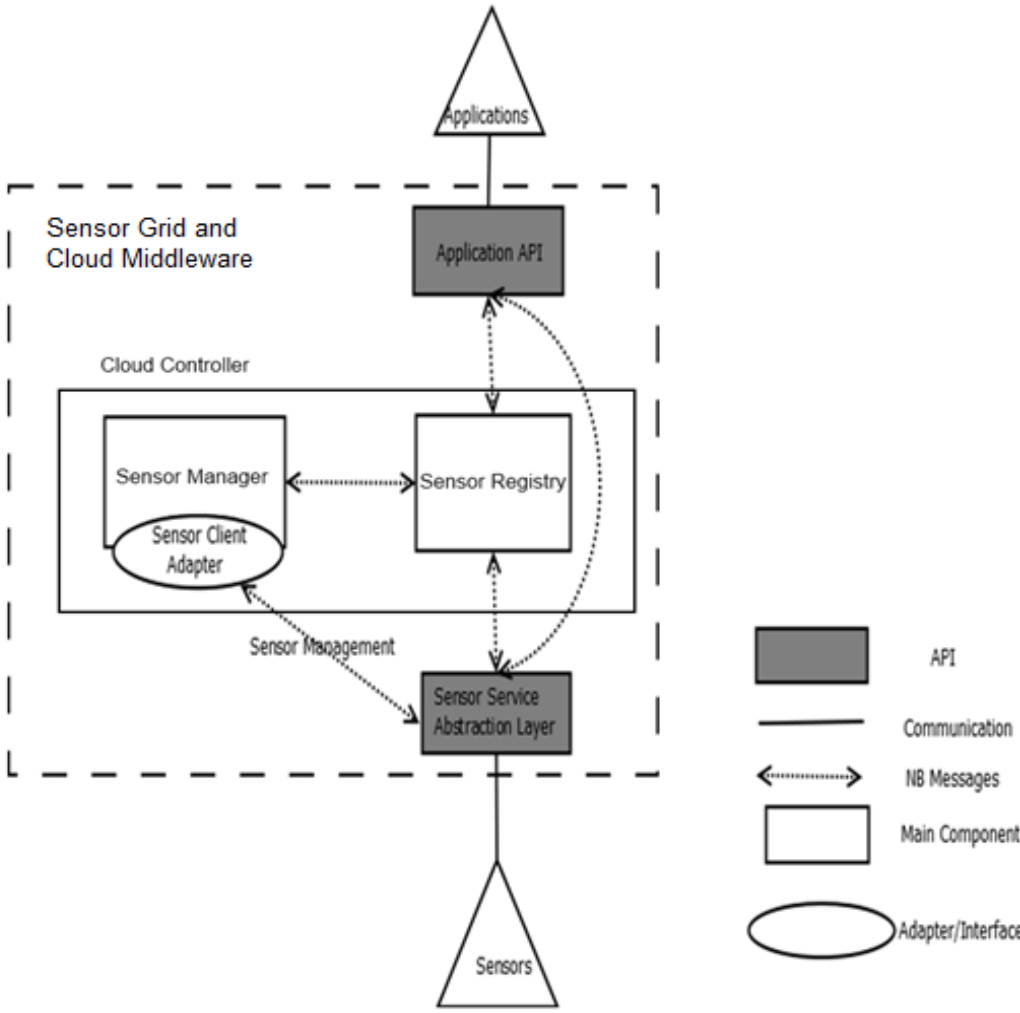
Indiana University

Our model of a "sensor cloud"



Sensor Cloud Middleware

- Sensors are deployed in Grid
- Sensors are discovered through the Sensor Cloud
- Grid and Sensor Grid are abstractions on top of the underlying Message Broker
- Sensors Applications connect vic simple Java API
- Web interfaces for video (Google WebM), GPS and Twitter sensors



Cloud4Sens

- A framework for managing sensing resources in the cloud and providing them as service
- Framework consists of two strategies to choose from,
  - Data Centric
    - Cloud provides sensed data to client without knowledge of how data is measured and processed
  - Device Centric
    - Clients use virtual sensing infrastructure and customizes it as per as their needs

Maria Fazio and Antonio Puliafito, IEEE Communications Magazine, Communications Standard Supplement, March 2015

Types of Services

- A cloud-oriented solution to integrate heterogeneous Monitoring Infrastructures (MIs), the consumer, into the cloud according to agreements between the Cloud4Sens provider and MI administrations
  - Real time access
  - Time scheduled access
  - Limited access to physical infrastructure or data

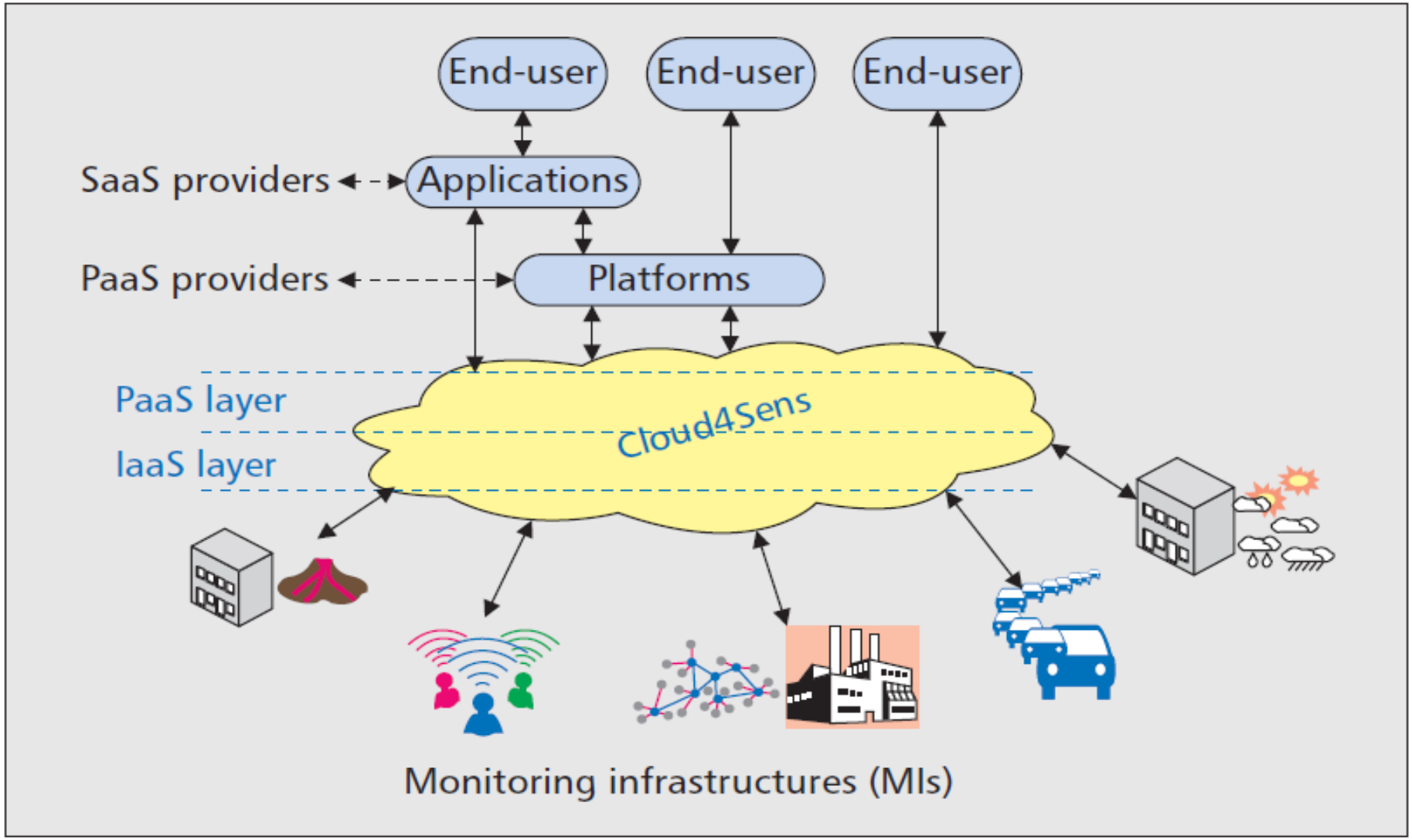
Different MI Models

- MI provides data as open sensory data through the Web.
- Cloud4Sens integrates such type of data and develops a specific adapter for such type of data provisioning
- MI owner is both the resource provider and consumer. They exploit Cloud4Sens to extend their physical infrastructure by using cloud virtual infrastructure

Cont..

- In Cloud4Sens, a client can be an end user or a software program.
- Cloud4Sens provides two types of services, at the IaaS and PaaS layers, according to the *data-centric* and *device-centric* models in offering sensing/actuation resources

Cloud4Sens: Architecture Overview



Data Centric Model

- Gathers physical measurements & environmental information from heterogeneous MI
- Organizes them according to a uniform format
- To support resource provisioning, Cloud4Sens implements a PaaS, storing and abstracting heterogeneous sensing/actuation data
- Clients do not need knowledge of monitoring system and can access data through high level interfaces

Device Centric Model

- Offers a sensing/actuation infrastructure to the clients, implemented as IaaS
- This infrastructure aggregates sensing and actuation resources that the applications exploit to deliver services to the end users
- Cloud principles enable the virtualization of such infrastructure, hiding specific deployment issues from the end users
- Resources are managed as a customizable virtual node

Summarize: Data-centric vs. Device-centric

#	Feature	Data-centric	Device-centric
1	Type of resources	Data	Infrastructure
2	Type of offered cloud service	PaaS	IaaS
3	Client needs	Environmental information, contextual measurements, and compound observations	Full control of of sensing/actuation resources, and the ability to customize the behavior of devices.
4	Support for heterogeneous distributed infrastructures	Yes	Yes
5	Need of abstraction technologies	Yes	Yes
6	Need of virtualization technologies	No	Yes
7	Decoupling between cloud and MI activities	High	Low

Mobile Cloud Computing and WSN

- Integration of WSNs and Mobile Cloud Computing (MCC) platforms
- WSN provide sensory data to the cloud platform
- Mobile users requests data from the cloud
- Addresses the areas of
  - Usefulness of obtained sensory data
  - Reliability of WSN

C. Zhu et. al. , Toward Offering More Useful Data Reliability to Mobile Cloud From Wireless Sensor Network, IEEE Transactions on Emerging Topics in Computing, Issue – 01, vol. 3, pp. 84 – 94, March 2015

Proposed Framework

- Proposes a WSN-MCC integration scheme called TPSS
- TPSS features
  - Time and priority based selective data transmission (TPSDT) for WSN gateway to cloud
  - Priority based sleep scheduling (PSS) algorithm to save energy consumption of WSN; gather and transmit data in more reliable way

Motivation

- Users maybe interested in this sensed data but, some sensed data will have more interest than other
  - Video data of storage room vs. front door, back door and windows w.r.t intrusion monitoring
- Not all sensory data useful/utilized
- Transmitting multimedia data can take up a lot of network bandwidth
- Data which is utilized more should be transmitted to the cloud with higher priority
- These data should also be gathered and transmitted reliably in order to conserve WSN energy

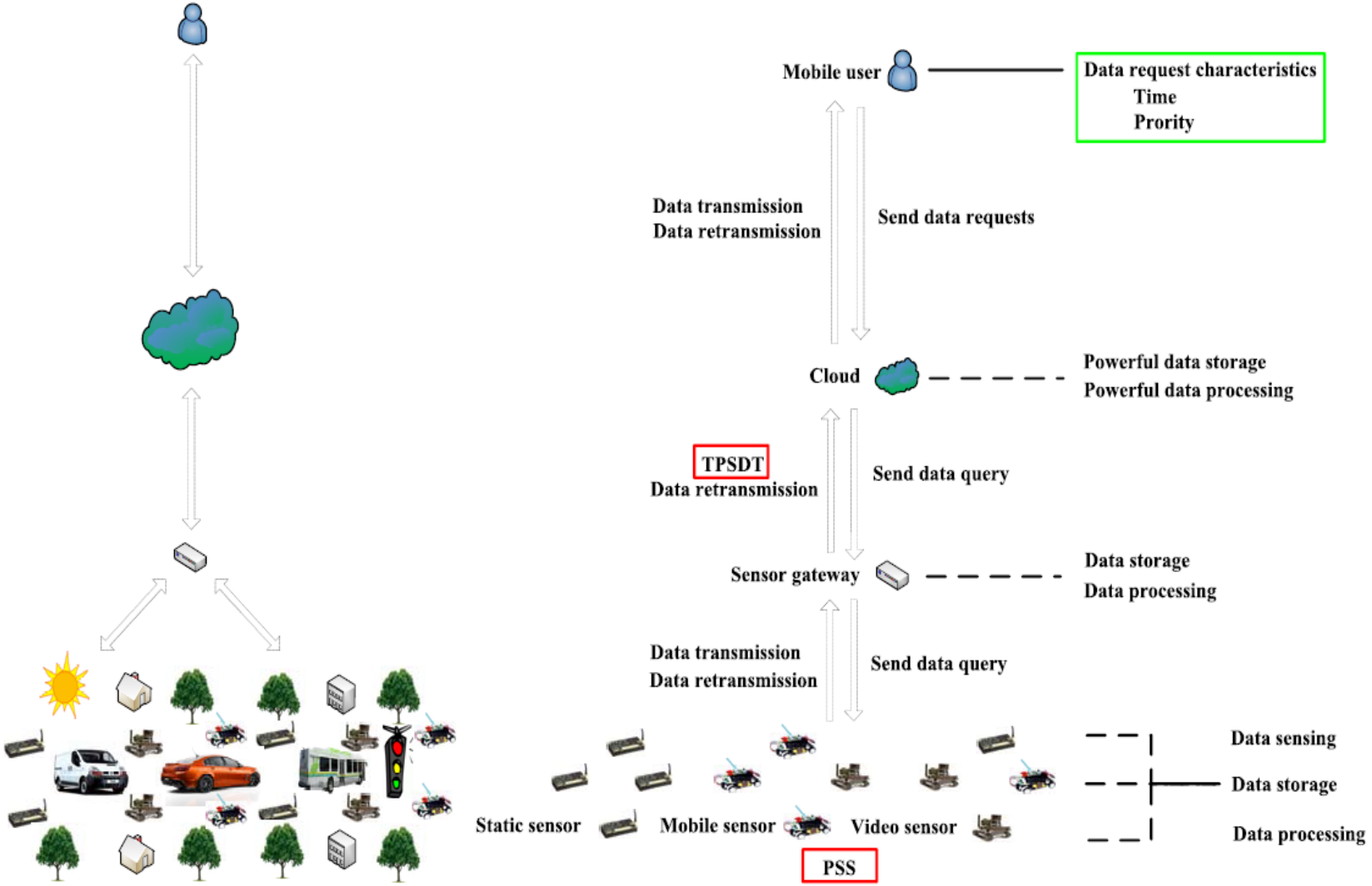
Usefulness of Sensory Data

- User data request features
  - Time: User request of certain types of data will be more frequent over a given time period of the day (Watch talk shows during lunch hours at work – 2pm to 3pm)
  - Priority: Traffic information of whole city is available, but a user will be more interested for information between their home and workplace

Reliability of WSN

- Reliability – whether WSN is continuously able to gather and transmit sensory data to cloud successfully
- Factors affecting reliability
  - Depletion of sensor energy: Limited/constrained sensor node energy
  - Failures in sensory data transmission : network congestion, limited bandwidth, and interference
  - Limitation in storage space of sensor nodes (Note: assumed sensors have sufficient storage space)

TPSS Schematic Overview



TPSDT

- Each gateway *g* sets timer recording current time
- For each time period *t*, each gateway *g* sends the sensory data to the cloud C, according to the start time and end time of *t* in the PTP table.
- For the transmitted data content, each gateway *g* sends the sensory data gathered by each sensor node in order, according to the priorities of probabilities
- For zero probabilities, data is not transmitted to the cloud
- **NOTE**: authors assume that cloud is able to analyze historical behaviors of mobile user data requests and then maintain a PTP table for each mobile user with respect to time and priority of each sensor node of interest.

Priority based sleep scheduling (PSS)

- A sleep scheduling algorithm that takes into account time and priority features of requested data
- Design Outline
  - Sensor nodes with non zero probabilities in PTP table should stay awake in a given time period
  - Whole sleep scheduled network should be connected such that it can transmit data to gateway node
  - A subset of all sensor nodes should be awake in in each time period to conserve energy

WSN-MCC Integration Outline Using TPSDT & PSS

- Sensor nodes determine their awake and asleep states with PSS
- Sensor nodes sense the environmental data with a set frequency and store the sensory data as well as process the sensory data
- Sensor nodes send the processed sensory data to the gateway *g* with the many to one and hop by hop pattern
- Gateway *g* stores the received sensory data and then processes the sensory data

- Gateway *g* selectively transmits the sensory data to the cloud C with TPSDT
- Cloud C further stores and processes the received sensory data
- If data transmission from *i* to *g* or *g* to C experiences data losses or failures, *i* or *g* performs data retransmission until the data transmission is successful
- Mobile user *u* issues data requests to cloud C and cloud C transmits the requested sensory data to the mobile user *u*

- If data transmission from C to *u* encounters data losses or failures, C performs data retransmission until the data transmission is successful
- Cloud C dynamically updates the PTP table with probability computations if the time and priority features of the requested data of the mobile user are changed and sends the updated PTP table to gateway in each time period *t*

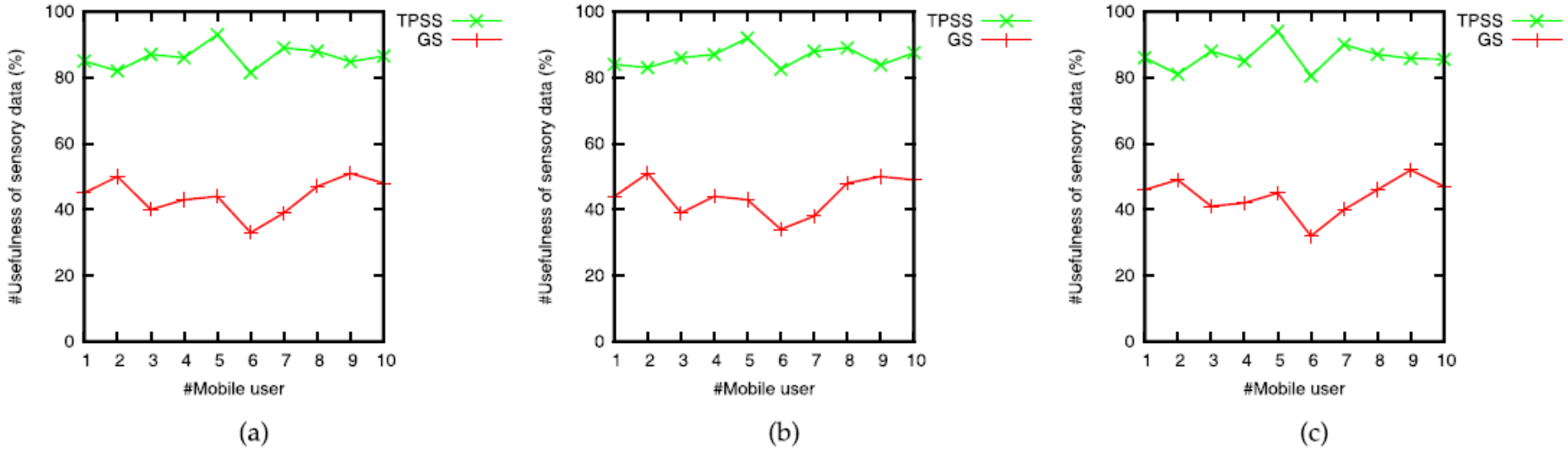
Evaluations

- Compared to a WSN-MCC integration scheme (GS or general scheme) which does not use TPSDT & PSS

Parameter	Parameter value
Number of clouds	1
Number of users	10
Number of WSNs	10
Number of sensor nodes	100
Number of gateways	1
Initial sensor energy	100000 mJ
Time period	1 hour
Network size	800×600 m^2
Default transmission radius	60 m
Transmission energy	0.0144 mJ
Reception energy	0.00576 mJ
Transmission amplifier energy	0.0288 nJ/m^2
Packet length	12 bytes
Number of packets	1000
k in PSS	1

Evaluation Parameters

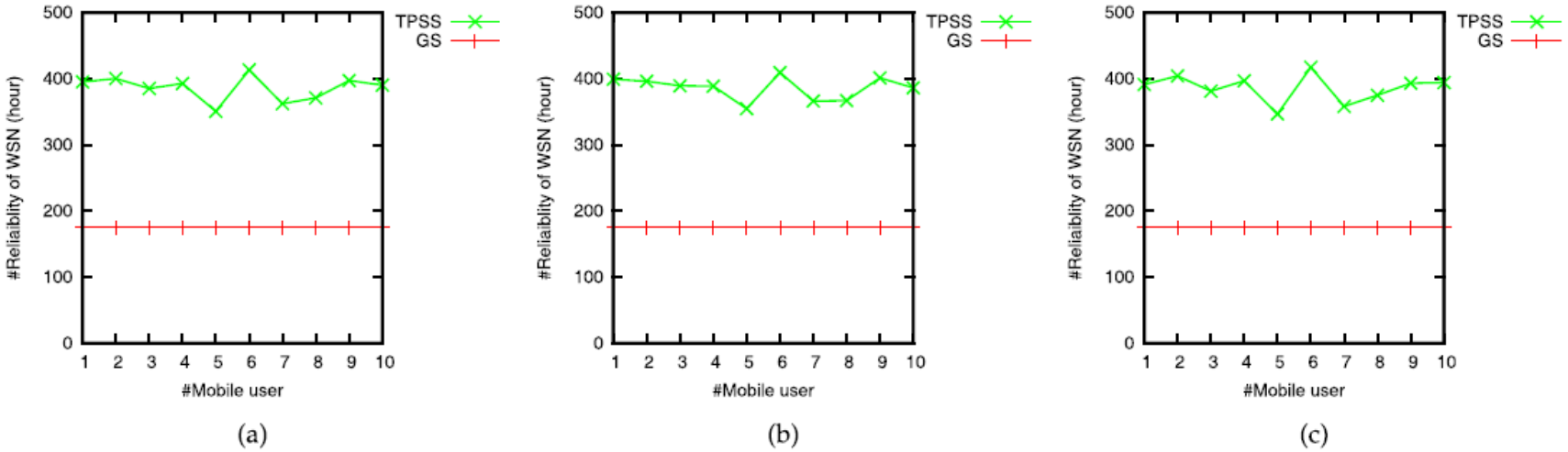
Results: Usefulness of Sensory Data



Average usefulness of sensory data for each mobile user over week 1 (a); in week 2 (b) and in week 3 (c).

Averaging over all mobile users, around 85% of the sensory data sent to the cloud with TPSS are useful to the mobile users, whereas only around 45% of the sensory data sent to the cloud with GS are useful to the mobile users.

Results: Reliability of WSN



Average reliability of WSN for each mobile user in week 1 (a); in week 2 (b) and in week 3 (c)

Over each of the three weeks observed, the reliability of WSN with GS is around 180 hours for all the mobile users, while the reliability of WSN with TPSS varies among different mobile users but averages to around 400 hours

Target Coverage using a Collaborative Platform for Sensor Cloud

- Sensor resources are provided as a service to the user
- Cost for offering such service depends on the number of physical sensors selected for covering the targets (Region of Interest)
- Proposed a two-phase algorithm to address the problem of selecting minimum number of physical sensors to cover a given territory

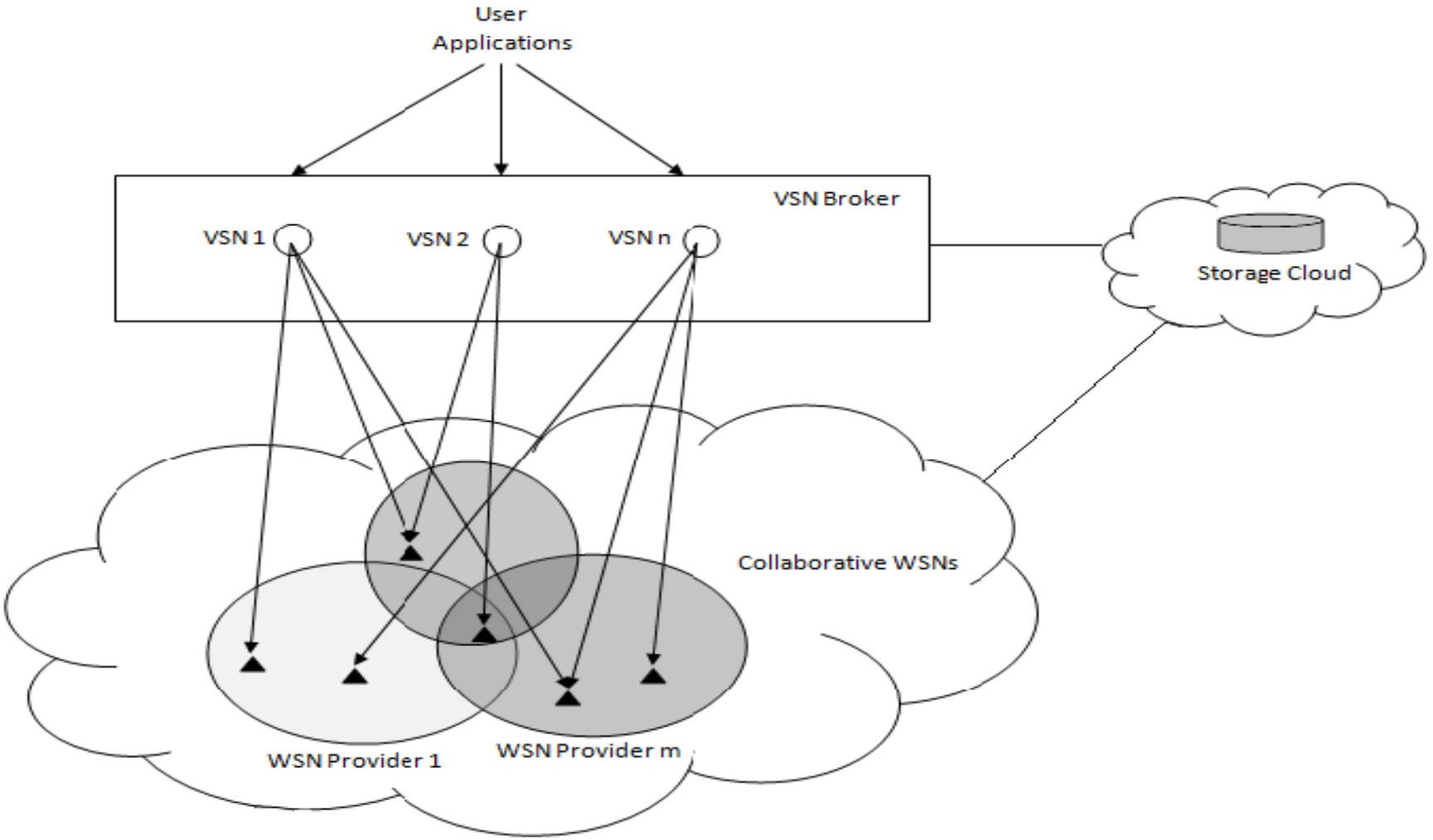
Biplab K. Sen, Sunirmal Khatua and Rajib K. Das, IEEE ANTS 2015

- First Phase: Coverage
  - Integer Programming Problem(IPP) is used to find a minimum set of sensors to cover the targets
- Second Phase: Connectivity
  - heuristic is used to extend the set such that connectivity of all the selected sensors is ensured

Sensor Cloud Architecture

- Sensor Virtualization
  - The partitioning of a physical sensor node into smaller virtual sensor nodes (VN) which sense and forward a physical parameter along with other collocated VNs
- Virtual Sensor Network (VSN)
  - Network of VNs over a dispersed geographical area, deployed on the physical sensor nodes of multiple WSN provides, for a particular sensor application that may change its sensing parameters as well as the area of interest on-demand

Proposed Architecture - Overview



Problem Formulation & Algorithm

- Given a set of applications, A_i , find the mapping where the i[th] application is mapped to j[th] VSN, such that the total set of physical sensors (TS) is minimized.

Formally

$$f: A_i \rightarrow (VSN)_j \; s.t.$$

$$TS = \cup_j \cup_{v \in VSN_j} (v.VN.SN)$$

Constraint 1: Coverage

- For each target $t \in \cup A_i.T$, there exists at least on sensors $s \in TS$ such that t is covered by s

Constraint 2: Connectivity

- For each sensor $s \in TS$, there is a path directly or via other sensors in TS to one of the base stations

Algorithm 1: Coverage

Find the minimum number of sensors required to satisfy all the applications, $A_i, 1 \leq i \leq n$

$T_{all} = \cup A_i.T$, set of all target region that needs to be covered

Define an array $y_j, 1 \leq j \leq m$ such that ,

$$y_j = 1 \; if \; t_j \in T_{all}$$
$$y_j = 0, otherwise$$

- $\{s_1, s_2, \ldots, s_n\}$ is the available sensors nodes and base stations ($\{s_1, s_2, \ldots, s_k\}$).
- A subset of sensors covers each region t_j if there is at least one sensor from the sensor subset that covers the given regions.
- IPP is used to find a cover consisting of minimum number of sensors

$$minimize: Z = \sum_{i=k+1}^{n} (x_i)$$

$$subject\ to: \sum_{i=k+1}^{n} (S_{i,j} \times x_i) \geq y_i, \forall j = 1, .., n$$

$S_{i,j} = 1$, if s_i covers t_j; 0 otherwise

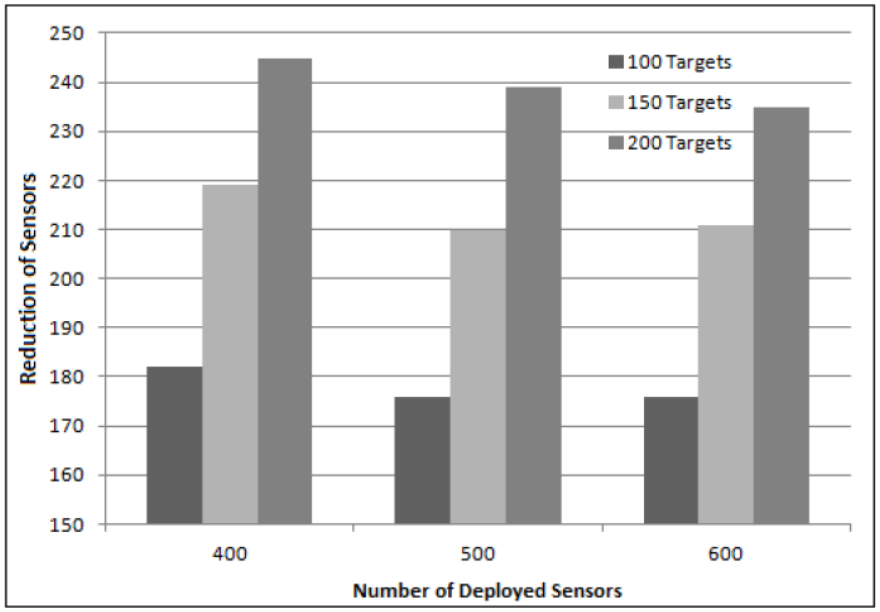
$x_i = 1$, if s_i is included in the cover; 0 otherise

Algorithm 2: Connectivity

- There has to be a path from each sensor in the cover to at least one BS. Find additional set of nodes that ensures this, called *connectivity nodes*

input : Adjacency Matrix adj, Set of Base Stations B, Set of Coverage Nodes cv
output: Set of Minimum Connecting Nodes M, Set of Disconnected Nodes D if the graph is disconnected
begin
  $M \leftarrow \phi$
  $(C_{BS}, C_{WBS}) \leftarrow$ findComponents (adj, B, cv)
  while $(C_{WBS} \neq \emptyset)$ **do**
    Find closest pair of components (c_1, c_2) s.t. $c_1 \subset C_{WBS}, c_2 \subset C_{BS} \cup C_{WBS}$
    if *no such pair found* **then**
      goto disconnected
    else
      $minPath \leftarrow$ set of nodes in the shortest path between c_1 and c_2
      $m_c \leftarrow c_1 \cup c_2 \cup minPath$
      replace c_2 by m_c
      remove c_1 from C_{WBS}
      $M \leftarrow M \cup minPath$
  disconnected: $D \leftarrow \cup_{(c \in C_{WBS})} c$

Results: Standalone vs. Collaborative



Collaboration Advantage over Standalone

No. of Targets	No. of Nodes	Stand-alone		Collaboration	
		cov	con	cov	con
100	400	153	116	61	25
	500	149	112	59	25
	600	147	112	57	26
150	400	201	112	76	19
	500	195	109	73	20
	600	190	111	70	20
200	400	245	106	89	16
	500	236	106	85	17
	600	230	105	82	19

Sensor node requirement:
Collaborative vs. Standalone

- Standalone: When an application is to be served by a single WSN
- Collaborative: When applications are served all available WSNs

Olympus: Building the Paradigm of CoS

- A CoS decouples the physical sensing infrastructure from users of sensing services
  - Automatic and dynamic (de)provision of resources according to applications demands
- Enables transparency of sensor types used
- Enables better sensor management
- Data from WSANs can be shared among users

Santos, I. L., Pirmez, L., Delicato, F. C., Khan, S. U., & Zomaya, A. Y. (2015). Olympus: The Cloud of Sensors. *IEEE Cloud Computing*, *2*(2), 48–56

Improving WSAN lifespan

- Processing and reducing the sensed data locally, within the physical WSAN improves the WSAN lifespan
- Information fusion consists of transforming / joining two or more pieces of information (data) from different sources, resulting in other information
- Virtual nodes are modeled as computational entities, each one capable of performing an information fusion technique

Improving the response time of applications

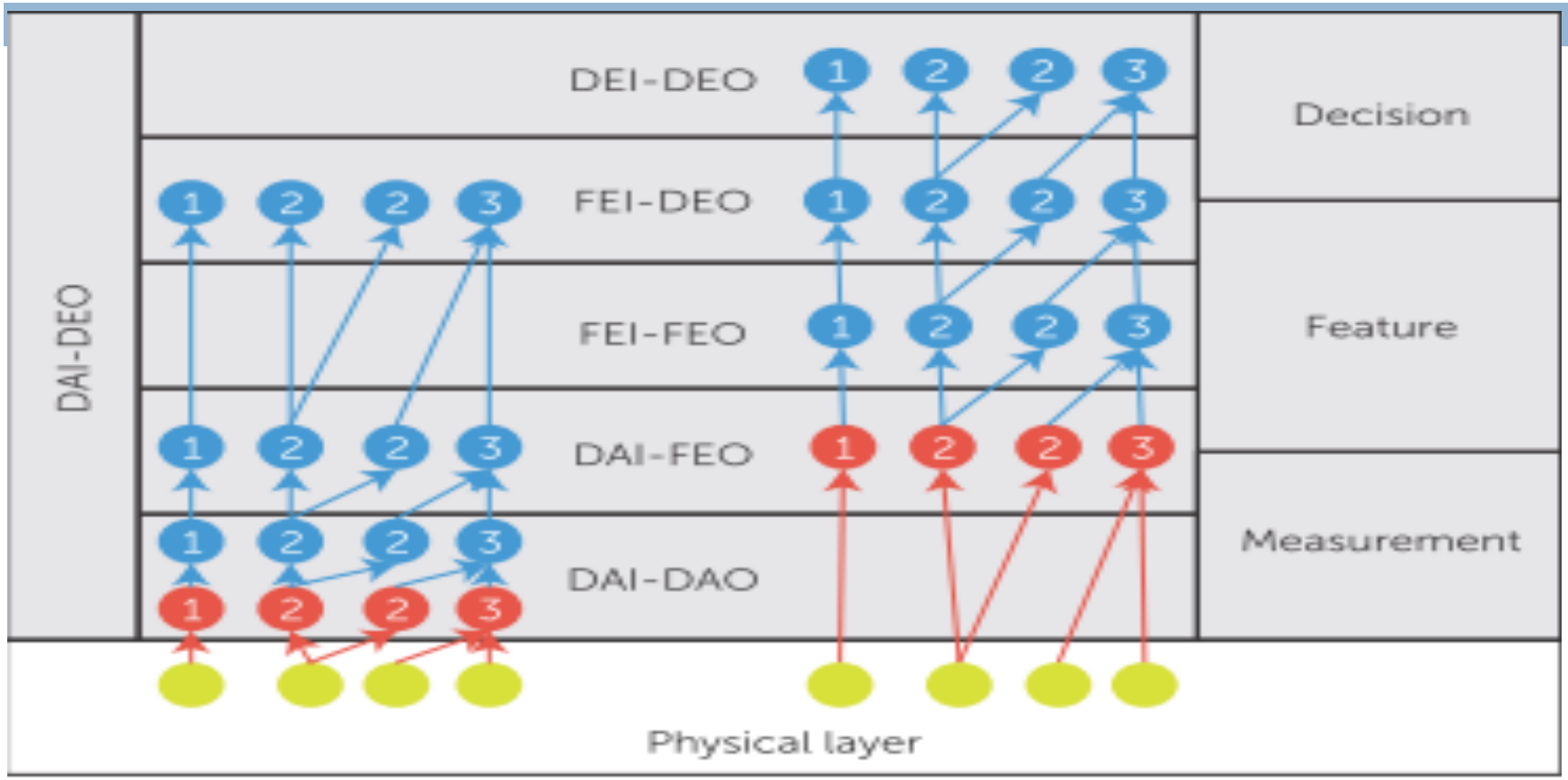
- In centralized CoS infrastructures, the response time of an application depends on several factors
  - the formation of communication bottlenecks close to the sink nodes
  - the size (in hops) of the physical WSAN
  - the delay in routing data inside the cloud
  - the delay in processing and making decisions within the cloud
  - the delay in issuing an actuation message back to the physical WSAN
- The data reduction approach lessens the time spent due to each of these factors, but does not eliminate them
- To overcome this restriction, a feasible approach is to decentralize applications' decision processes, leveraging the nodes' in-network processing capabilities

Olympus

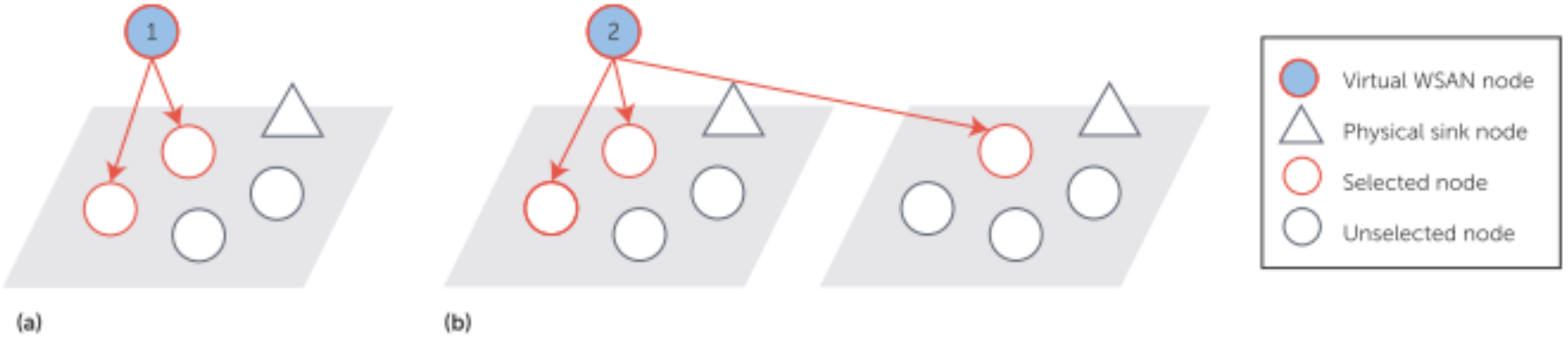
- Olympus is an information fusion and CoS based decentralized WSAN virtualization model
  - Uses information fusion to reduce and provide data at a given abstraction level required by users
  - It's a decentralized WSAN virtualization model, in which physical nodes can create and run the virtual nodes locally

- Olympus supports both centralized and decentralized applications

Linking virtual nodes and data abstraction levels of information fusion



Examples of many-to-one virtualization



- (a) within the same physical wireless sensor and actuator network (WSAN) and
- (b) comprising different physical WSANs.

Security Risks in Sensor-Cloud

Proposed Thrust Areas

The broad motivation of this project is an investigation of the security issues and challenges in the sensor cloud environment, and to design approaches to secure emerging sensor-clouds against a variety of attacks.

- **Secure Pre-deployment: D**ealing with security issues prior to the deployment of the sensor-cloud:
- **Secure Pre-processing: D**ealing with security issues prior to the execution of sensor-cloud missions
- **Secure Runtime: D**ealing with security issues during runtime. In particular, trust and key management issues

Risks in Developing Sensor-Cloud

Problems

Despite the benefits, sensor-cloud computing emerges with a unique opportunity for abuses and attacks.

- Sensor nodes are susceptible to attacks including node capturing and compromising
- Wireless communications can be eavesdropped, captured, or tampered
- The infrastructure of the sensor-cloud can be misused by malicious users
- The virtual network topology can be very different from the heterogeneous physical topologies underneath, complicating security
- In-transit data due to the on-demand operations are unattended and processed in near real-time. Hence, it is difficult to foresee potential security violations in advance
- Most security best-practices cannot be used in WSNs due to the limitation in communication and computation capability

Secure Code Dissemination in Sensor Clouds

- Wireless sensors in sensor clouds are dynamically provisioned.
  - New users need to install new applications
  - New applications need code dissemination from the base station, which is forwarded by nodes.
  - Forwarding packets frequently consumes energy, the code dissemination should be efficient.
- Clusters of wireless sensors in a sensor cloud may be owned and used by many different entities
  - Code is forwarded by these nodes: Confidentiality of code is required.
  - Integrity of the code is required

Madria, et al., Secure Code Dissemination in Sensor Clouds **IEEE MDM 2014**

Overview of the Approach: Security Challenges

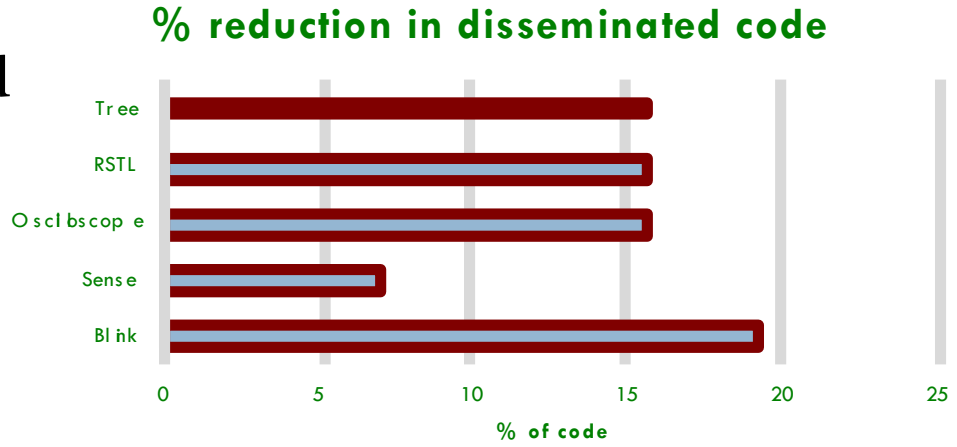
- When functions are stored on the sensor nodes, requesting specific functions would leak information.
  - Use Proxy Re-encryption to encrypt the code stored on the sensors to provide confidentiality.
  - To prevent nodes from misusing the encryption key, Proxy Re-encryption is used

- Malicious sensor nodes may try to tamper the code stored on them so as to change the functionality of the code.
  - To prevent code storing nodes from injecting corrupt code, Bloom Filter is used.

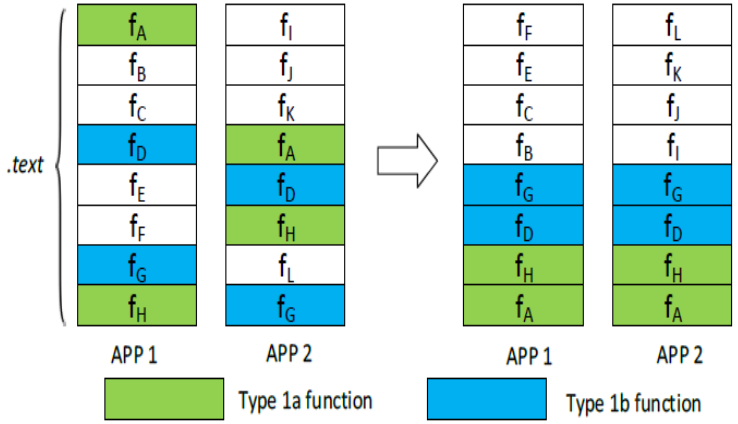
Proposed Solutions

- Restructure the application codes to find common functions across the applications and distribute these functions *a priori* into the network.
- New application code consists only of new functions, while common functions can be picked up from the network, thus reducing overall wireless communication.
- Usage of symmetric proxy re-encryption scheme to store, the common functions, encrypted on sensors, such that they can be re-encrypted with re-enc key without revealing the enc/dec key.

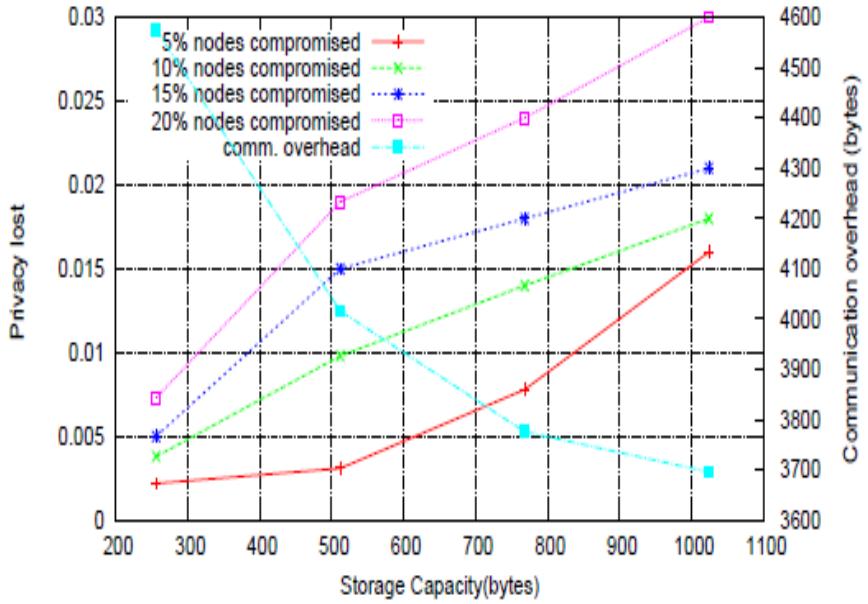
- Our implementation shows upto 19.06% reduction in code compared to Seluge.

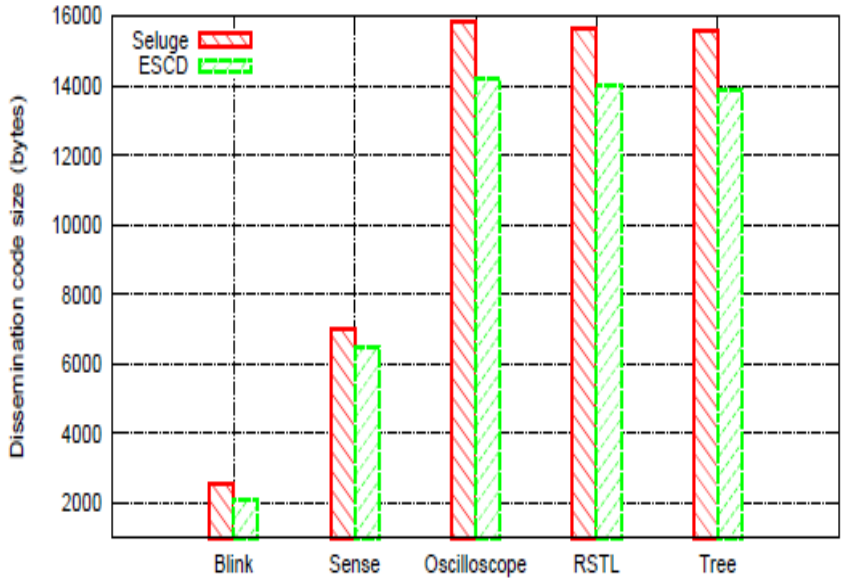


Efficient & Secure Code Dissemination in Sensor Clouds

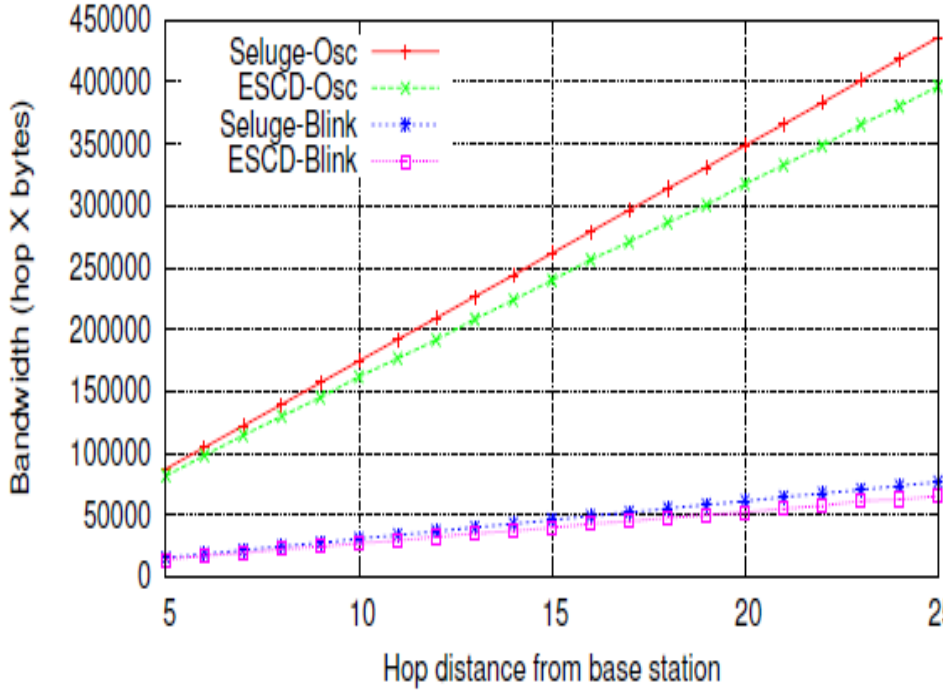


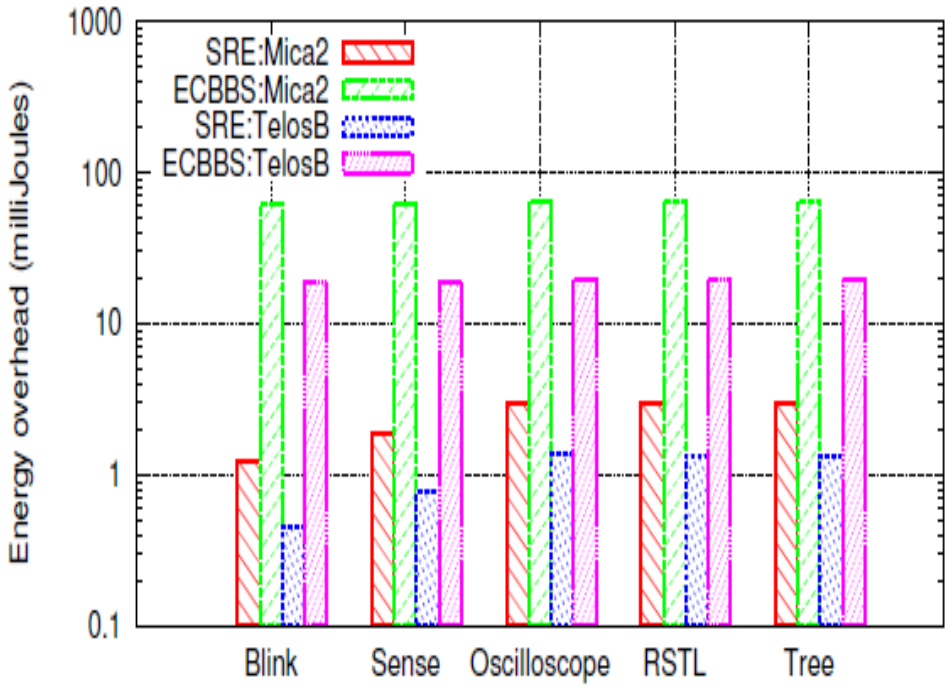
- A large amount of code is common between applications in WSNs
- In our approach the code common b/w the applications is loaded onto the sensors. In subsequent code disseminations this common code can be picked up by nearby sensors, thus reducing the overall code transmission
- Our security scheme based on proxy-re-encryption and bloom filters provides confidentiality and integrity of code.





Performance Analysis





Bandwidth Consumption compared to Seluge. The difference would be greater for large applications and large networks

Energy overhead of the scheme compared to Seluge. (Seluge does not provide confidentiality of the packets)

Privacy and Integrity Preserving Data Aggregation in Sensor Cloud

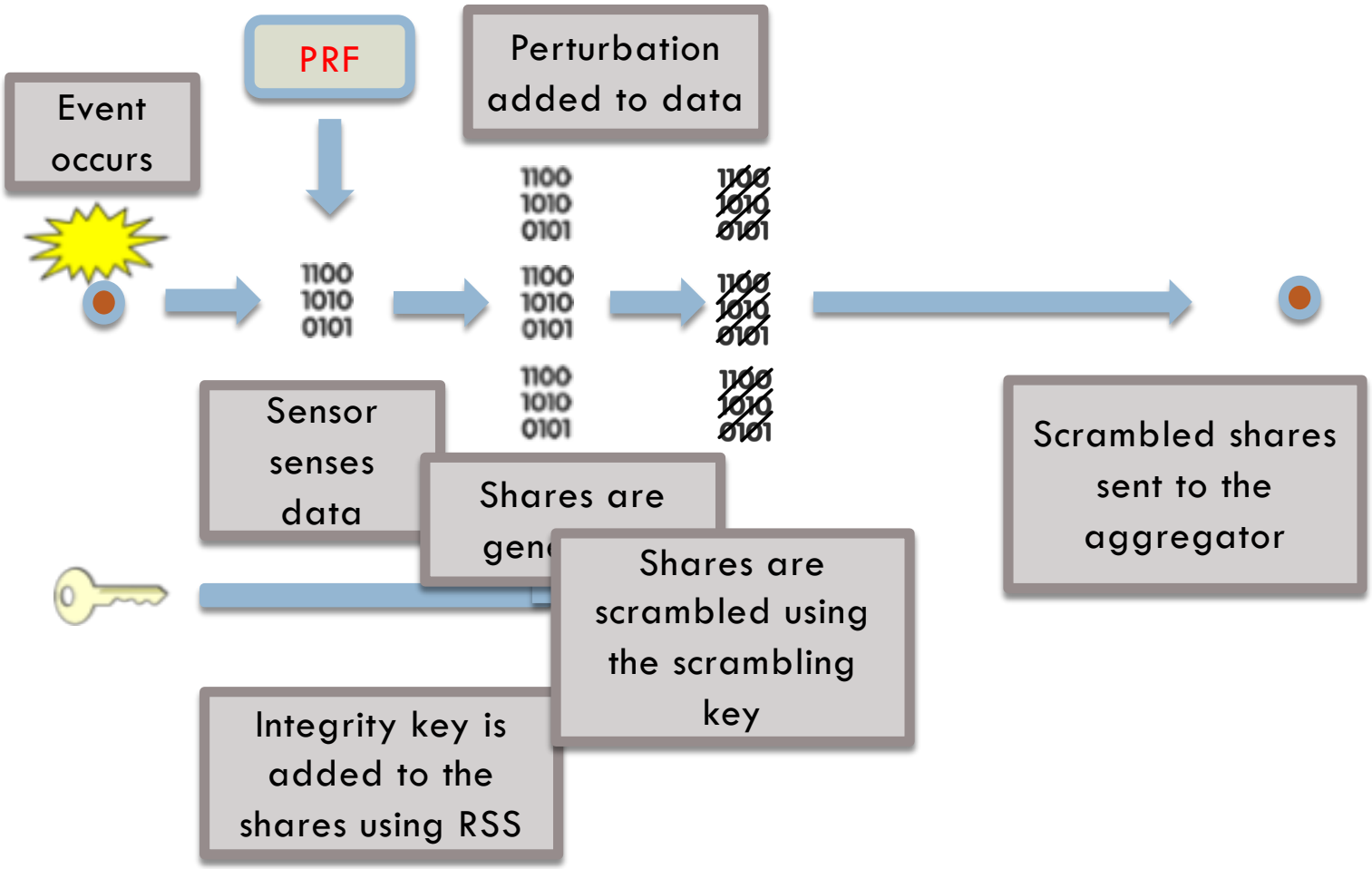
- Develop a data aggregation scheme which aggregates sensor data privately without having the individual sensors reveal actual data
- Develop an integrity preserving scheme which can detect data injection attacks by corrupt sensors
- Provide a scheme to localize the malicious nodes in the network
- Simulate, implement and test the data aggregation scheme in comparison with others on a hardware platform using Telos Motes

Madria, et al., Privacy and Integrity Preserving Data Aggregation in Sensor Cloud, IEEE SRDS 2015

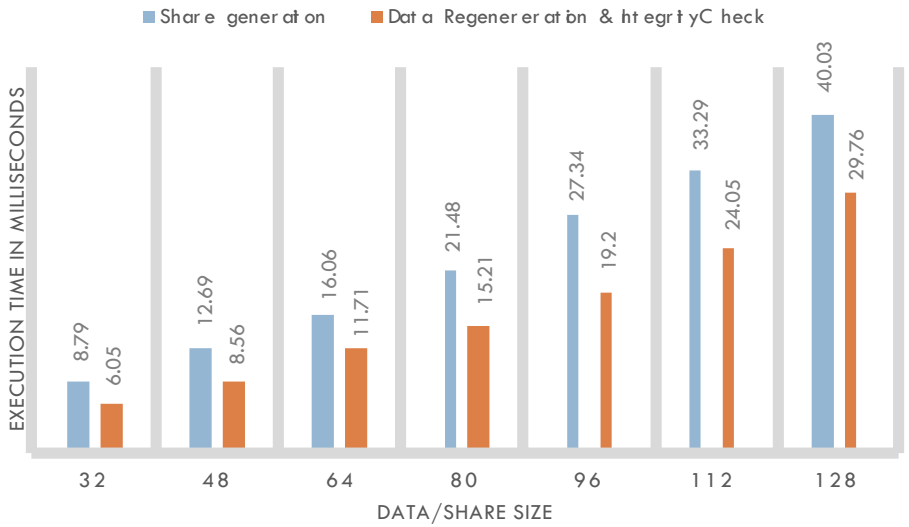
Proposed Approach

- Use recursive secret sharing as a data structure to hold the data and the integrity key
- Since secret sharing is homomorphic, both the data and the integrity key are automatically aggregated
- Integrity key is constructed using the data and keying material shared with the BS
- Use scrambling keys to scramble the shares (perturbation)

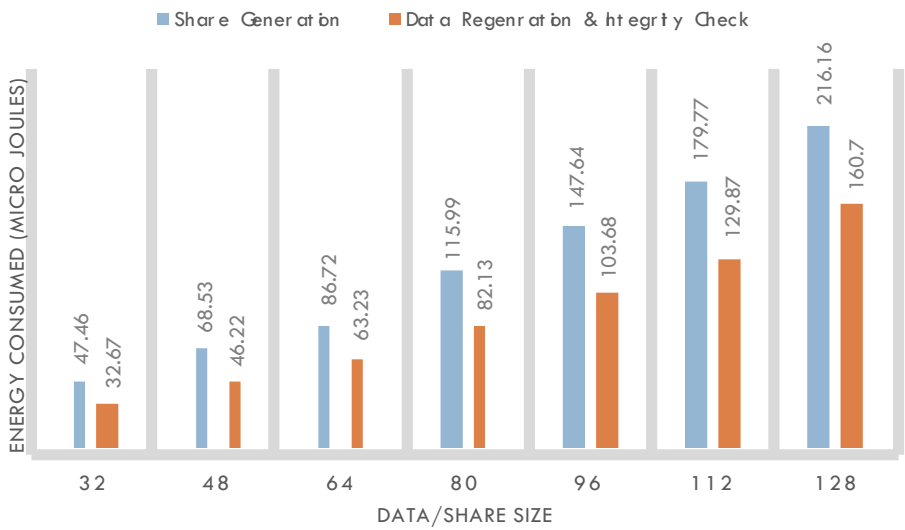
Proposed Approach



Performance Analysis



Execution Time

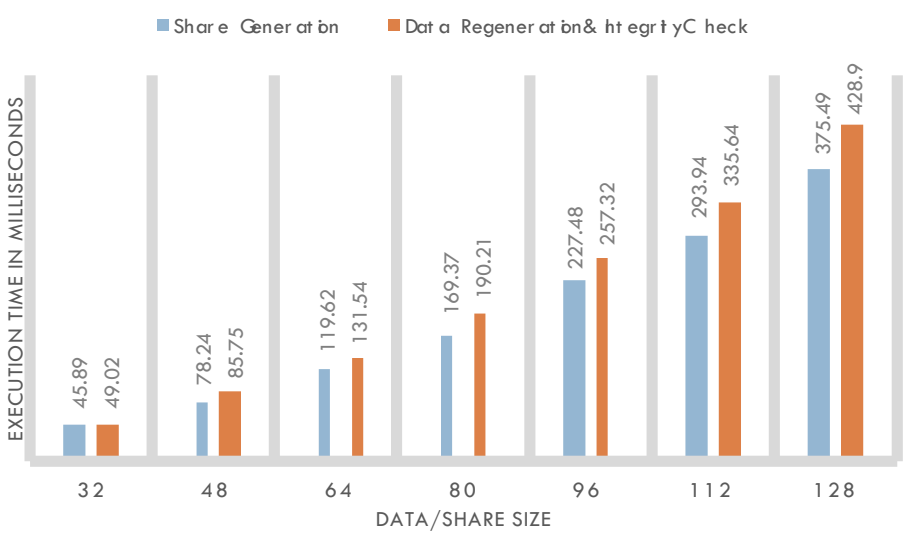


Energy Consumption

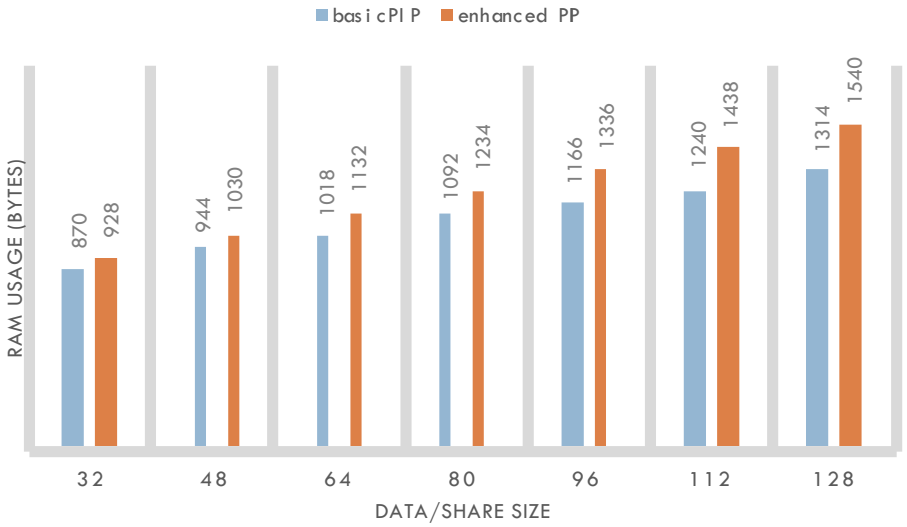
Scheme	32/128 bit	48/192 bit
SDA	79.77 mJ	115.7 mJ
PIP	0.047 mJ	0.068 mJ

Implementation of Basic PIP on TelosB motes

Performance Analysis

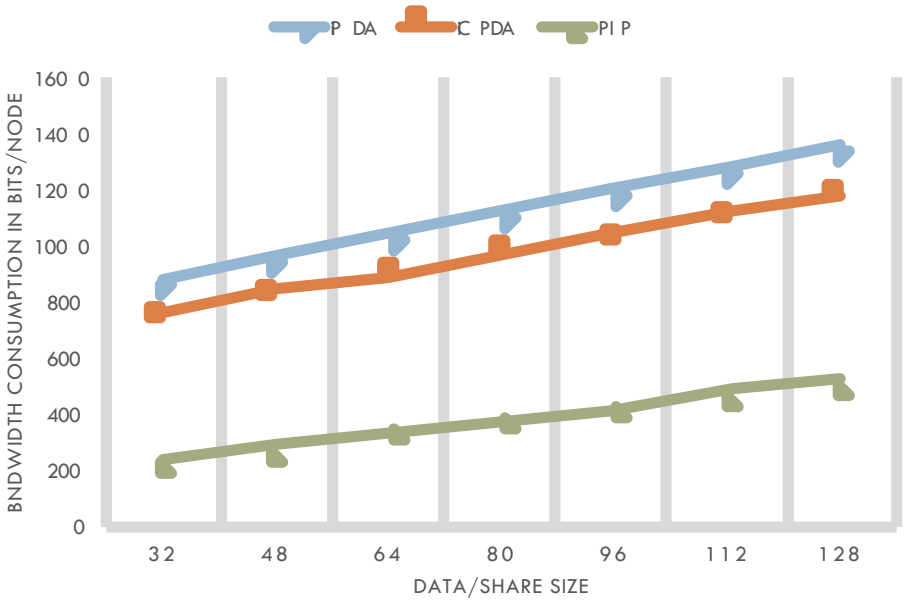


Execution Time of enhanced PIP on TelosB motes

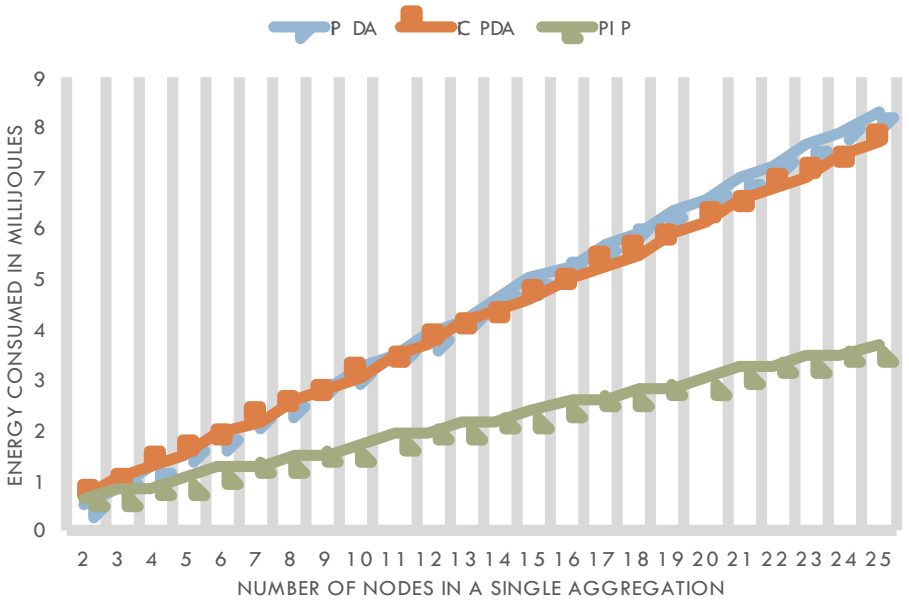


RAM usage of basic and enhanced PIP on TelosB motes

Performance Analysis



Bandwidth consumption per node on iPDA, iCPDA and PIP for varying share size.



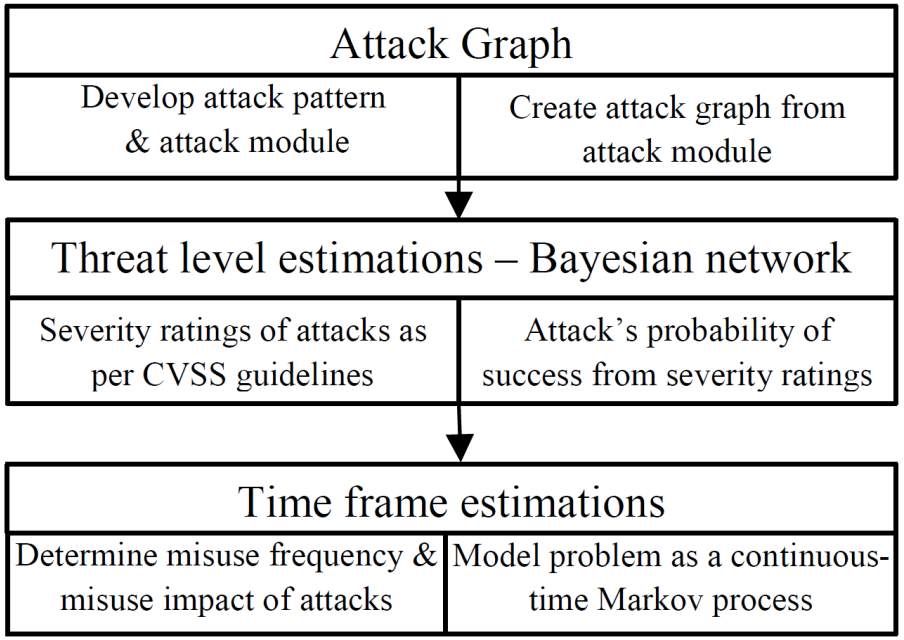
Energy consumed at each node as a function of number of nodes involved in aggregation.

Risk Assessment and Mitigation in Sensor Clouds (Madria et al., IEEE TSC, 2016)

- Wireless Sensor Networks (WSNs) in a Sensor Cloud are deployed in an ad-hoc fashion across a large geographical region unattended
- Resource constrained which impedes the incorporation of desired security measures to patch up all the security risks that might be present
- Networks are integrated with the cloud platform which increases their vulnerability to security attacks
- Risk assessment for such networks is a pre-requisite before efficient and effective security measures can be formulated
- One must be able to take into account the logical co-relation between different feasible attacks.
- For example, a malware propagation from the cloud side to the WSNs can lead to node subversion. The subverted node can then be used to execute attacks like Sybil and Sinkhole

Proposed Framework

- Proposed a Risk Assessment Framework for WSNs in a Sensor Cloud using Attack Graphs consists of the following three modules:



Results: Attack Graphs

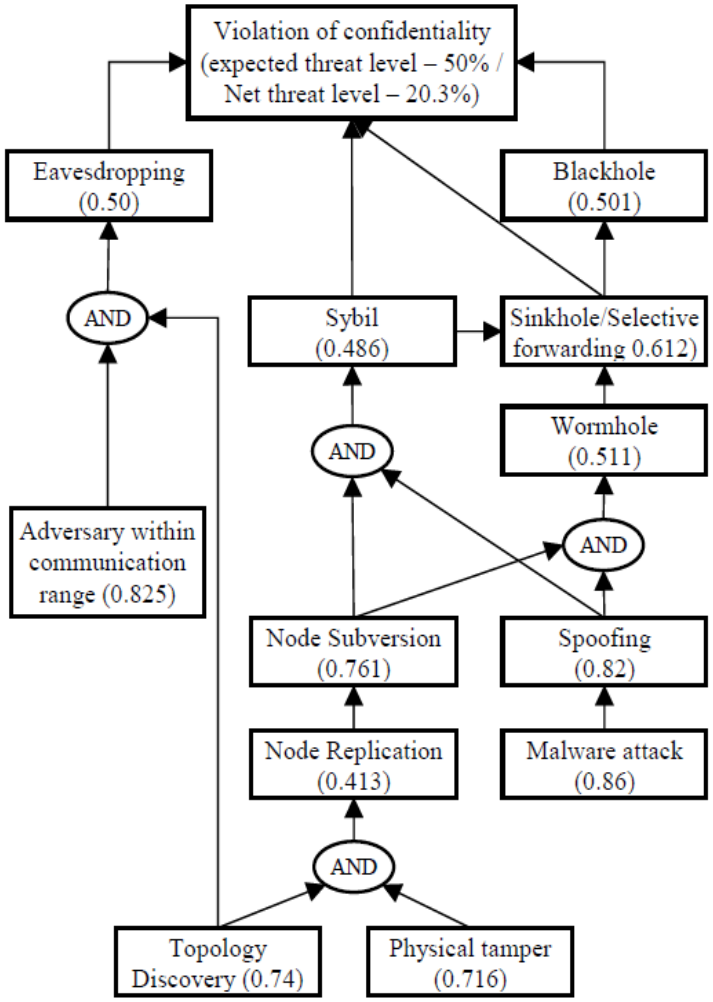


Fig. 1: Attack Graph with Violation of Confidentiality as the root node. Each attack node is accompanied by its probability of success in absence of security measures

- Attack graphs are generated by studying the attack pattern of an attack and the network conditions required to execute them as well the network conditions arising due to their successful execution.
- Attack graphs are represented as Bayesian networks.
- Nodes are assigned with the probability of success of an attack which is computed using their Misuse Frequency and Misuse Impact.

Results: Time Frame Estimations

- Time frame estimations predict a time period depicting the degradation of WSNs security parameters – confidentiality, integrity, and availability.
- Service levels are formed using the impact level of different feasible attacks and the transition matrix is computed using an attack's frequency of execution

Service levels	Attacks	Attack Pattern
SL0(0.0)	-	-
SL1(0.14)	Node Subversion, Spoofing, Node Replication, Malware attack, Wormhole, Selective Forwarding	C; I
SL2(0.33)	Eavesdropping, Sybil, Selective Forwarding, Spoofing, Alter/Replay, Acknowledgment Spoofing, Node Malfunction	C; I
SL3(0.50)	Frequency Jamming, Denial of Service	A
SLx(1.0)	-	-

Fig.2: Service Level generation for different feasible attacks on a WSNs. The numbers depict the impact level of all attacks belonging to a service level

	SL0	SL1	SL2	SL3	SLx
SL0	0	$(0.83)_{C,I,A}$	$(0.81)_{C,I}$ $(0.77)_{A}$	$(0.83)_{A}$	0
SL1	0	$(0.83)_{A}$	$(0.66)_{C,I}$ $(0.64)_{A}$	$(0.69)_{A}$	0
SL2	0	0	0	$(0.44)_{A}$	$(0.44)_{C,I}$
SL3	0	0	0	0	$(0.20)_{A}$
SLx	0	0	0	0	0

Fig.3: Generation of transition matrix using Misuse Frequency of different attacks.

From Fig.3, estimation of 0.20 signifies that if a time frame of 12 months is assumed, the availability of a WSN might affected beyond the point of repair in a time period of 3 months (20% of 12 months). Hence it is advisable to perform maintenance every 2 months, if possible.

Access Control of Data in Sensor Cloud

- The security issue we tackle in this paper is user access control of sensor data

  Only an authorized user should be able to access sensor data and the user should only access data he/she is authorized to

1. User does not have direct access to sensors and the sensors may be owned by other parties
2. Which sensors are used and how many sensors are used is not known in advance
3. Sensor data is aggregated in network
4. The sensor owner may want to place restrictions on the who accesses their data

Distributed Attribute Based Access Control of Aggregated Data in Sensor Clouds, in IEEE proceedings in 34th Symposium on Reliable Distributed Systems (SRDS 2015) Montreal, Canada. **IEEE Best Paper Award**

Our contributions

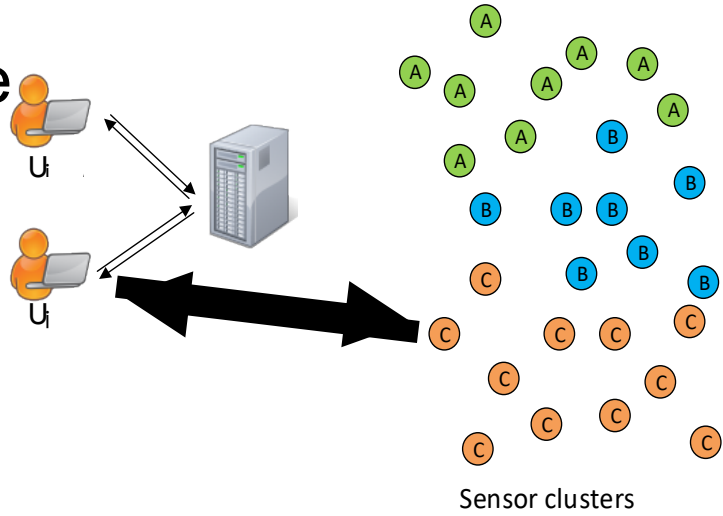
- A distributed, fine grained access control scheme for aggregated data in sensor clouds
- A scheme which considers runtime modification of authorizations on sensor data
- Integration of access control scheme with a secure data aggregation scheme

Wireless Sensor data is often collected as aggregate of the network data, therefore it is important that any access control scheme for wireless sensors, work with a secure data aggregation algorithm

System Model

- The sensor cloud consists of three sets of entities, the Users (U), the Sensor Cloud Administrator (SCA) and the Sensor Nodes (SN)
- To access data from the sensors, a user first contacts the sensor cloud administrator (SCA) with a query based on the attributes of the data.

- SCA returns a secret key over these attributes to the user.
- This secret key is then used to access data



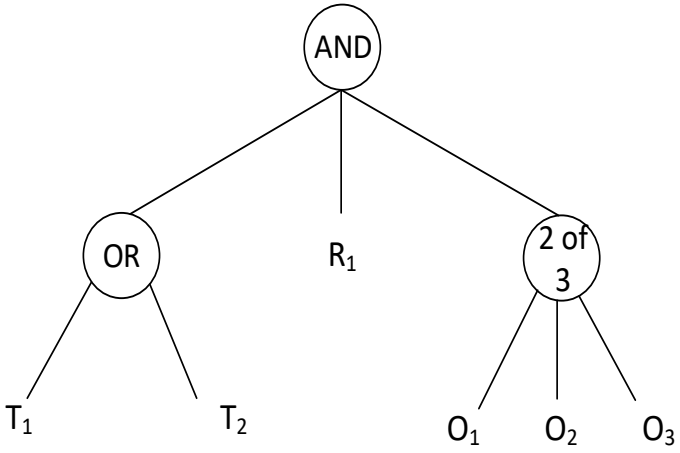
Adversary model

- The adversary in this system is a user who wants to access data for which he/she has not received access authorization from the SCA.

- The goals of the adversary are
  - To get access to the data, for which the adversary does not have the secret key
  - To access data for which the adversary does not have authorization
  - To tamper with the keys and data meant for other users, so as to disrupt the protocol

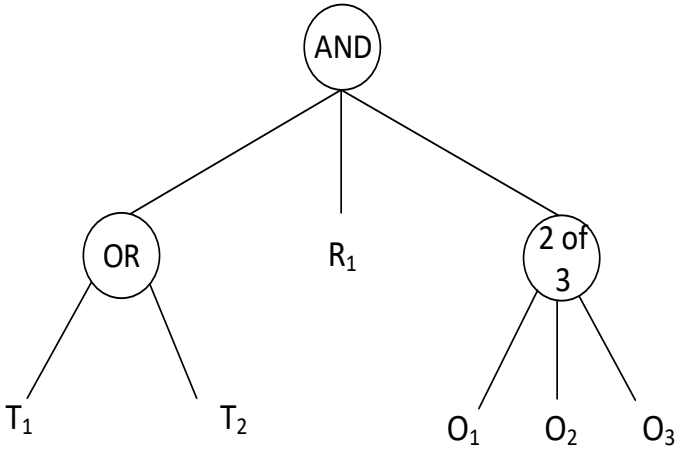
User Queries as Access Trees

- User queries for data can be implemented as access trees
- Example:
  - Data which is either of sensor type T1 or T2, from sensors which belong to any two of the three sensor owners O1, O2 and O3 and which are deployed in region R1



KP-ABE(key policy attribute based encryption)

- We start with KP-ABE for fine grained access control
- Using KP-ABE one can generate a secret key over an access tree which will decrypt data only if it satisfies the attributes in the access tree along with the structure of the tree



KP-ABE(key policy attribute based encryption)

- The sensors are loaded with the public key PK
- When a user wants to access some data, it creates a query based on the attributes of the intended data.
- This query is used to create an access tree, which is then further used by the SCA to create the secret key.
- When the query is disseminated in the network, the sensors check whether they satisfy the query based on their attributes.

KP-ABE(key policy attribute based encryption)

- If a sensor satisfies the query in the access tree, it generates a session key which is encrypted by using the PK fragments of the attributes.

$Enc = (K*Y^s, \{E_k = sT_k\}$ for all k where k is the attributes in the query), K is the session key

Where s is a random number chosen by the sensor

- This ciphertext Enc is given to the user. The user can decrypt this data if he/she has the secret key derived from the access tree otherwise not

KP-ABE decryption

- Decryption takes place according to the access tree in a bottom up manner

- KP-ABE and aggregation

We want to aggregate encrypted data but KP-ABE is not homomorphic and so is not suitable to ciphertext addition

We encrypt data first by Paillier encryption and then use KP-ABE to encrypt it further based on the attributes

Putting it all together:

Generate and encrypt random session keys

1. The user sends its query to the SCA
2. The SCA generates an access tree (γ) over the query and creates a secret key over this tree.
3. The SCA returns the secret key and (γ) to the user
4. The user disseminates (γ) as the query in the sensor network
5. A query tree is formed. The tree consists of nodes, which satisfy the user query
6. Once the tree is formed, each node generates a random session key and encrypts it using Paillier encryption
7. The Paillier encrypted ciphertext is again encrypted using KP-ABE
8. This ciphertext is aggregated on the intermediate nodes, on its way to the user

Putting it all together:

Aggregate and transmit session keys to the user

1. This encrypted session key gets aggregated in the network.
2. The user receives the aggregated ciphertext. If the user has received a corresponding secret key from the SCA, it will be able to decrypt the session key other wise not.
3. Once the user has decrypted the session key, the user will generate PIP[1] keys.

[1]PIP was published in SRDS 2013

Putting it all together:

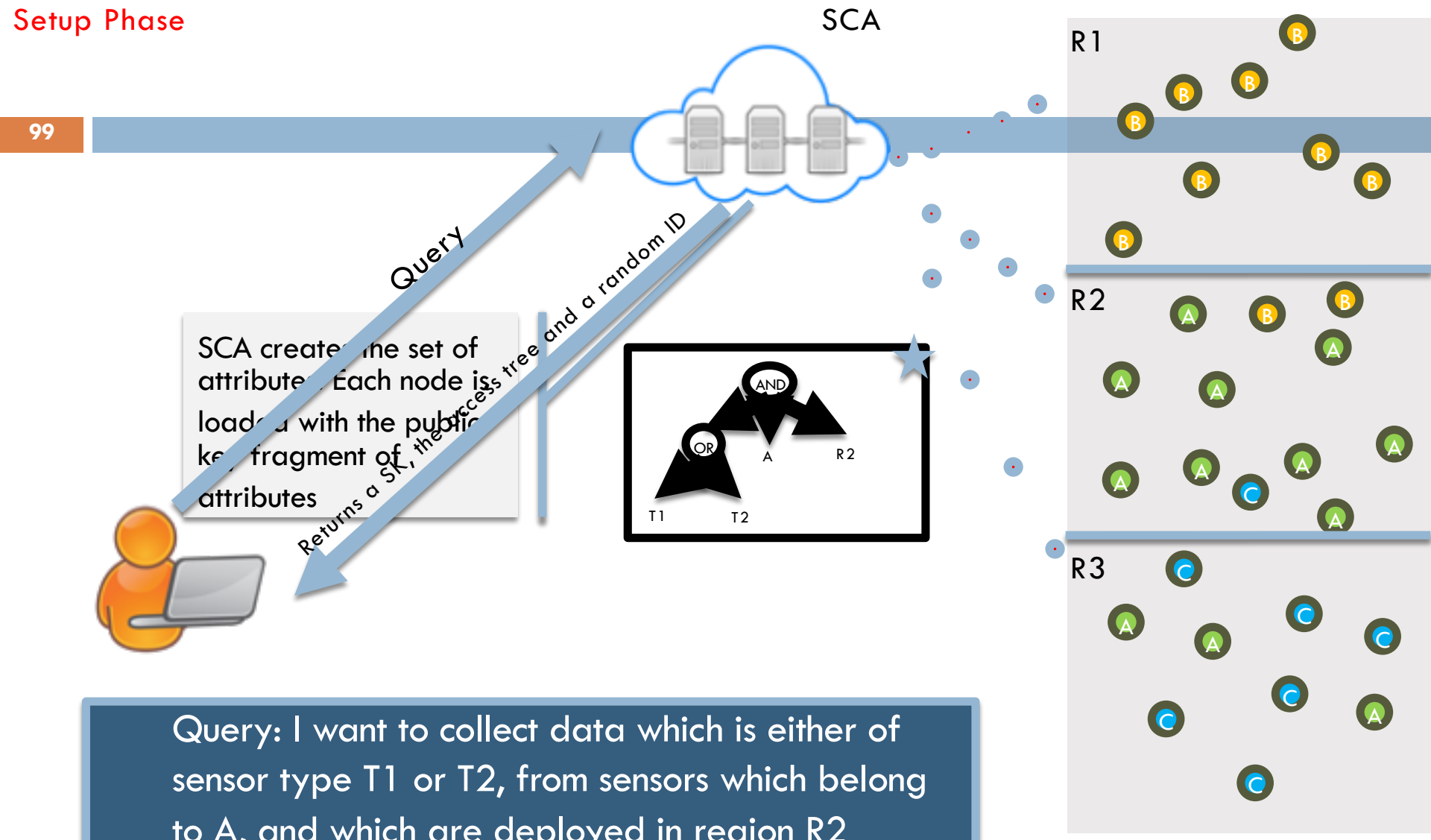
Generate PIP keys from the session key for data collection

1. PIP is a secure data aggregation algorithm which works on symmetric keys and provides integrity and privacy of data
2. The sensors on their end can generate PIP keys from their session keys.
3. Once the keys have been established, the sensors use these PIP keys to encrypt and transmit data securely to the user
4. The user on the other end can decrypt PIP encrypted data with the aggregate PIP keys

PIP

- PIP is a secure data aggregation scheme which uses symmetric keys
- It provides integrity and privacy of aggregate data
- It uses a recursive secret sharing scheme based on Shamir's secret sharing.
- PIP also provides a scheme to locate malicious nodes in the network which inject false data in the aggregate
- PIP uses a secure tree algorithm adapts itself to work around nodes which are either dead or maliciously stop sending data forward

Setup Phase

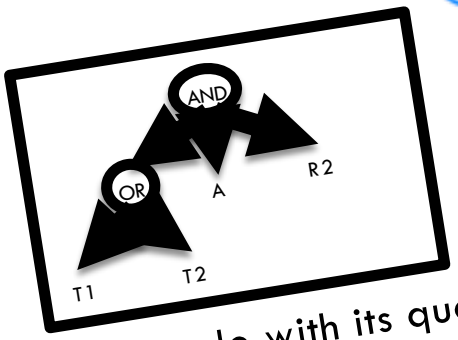


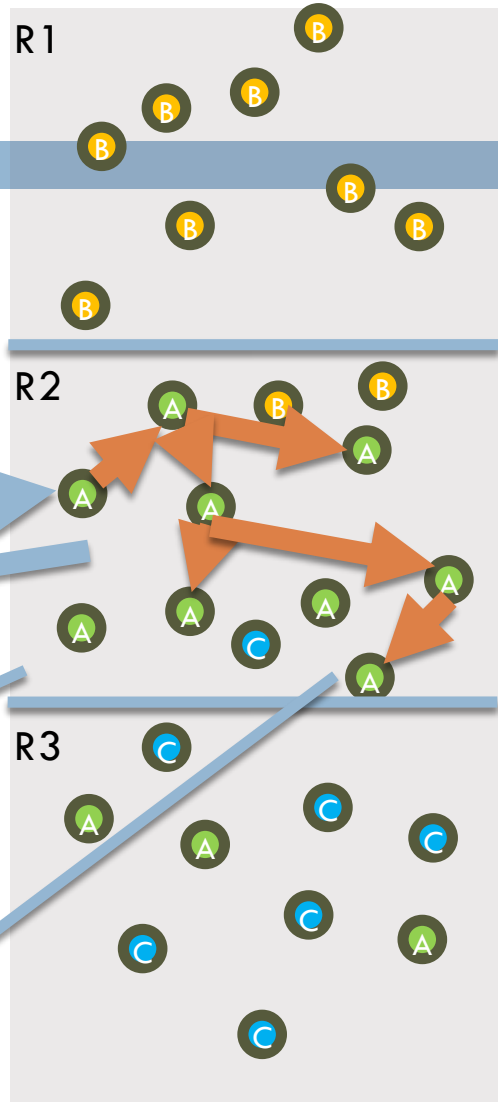
Query: I want to collect data which is either of sensor type T1 or T2, from sensors which belong to A, and which are deployed in region R2

Some nodes such as ID are not shown for the sake of simplicity

Data Aggregation key generation Phase

SCA





User contacts any node with its query (access tree Y)

User receives an encrypted aggregated session key $\Sigma Enc = (\Pi C_p Y^{\Sigma s}, \{E_k = \Sigma s T_k\}\ k \in Y$ which can be decrypted only by the SK it received from SCA earlier

A query tree is formed

Each node generates a partial session key s and encrypts it as $Enc = (C_p Y^s, \{E_k = sT_k\}\ k \in Y$ which is aggregated and sent back along the tree

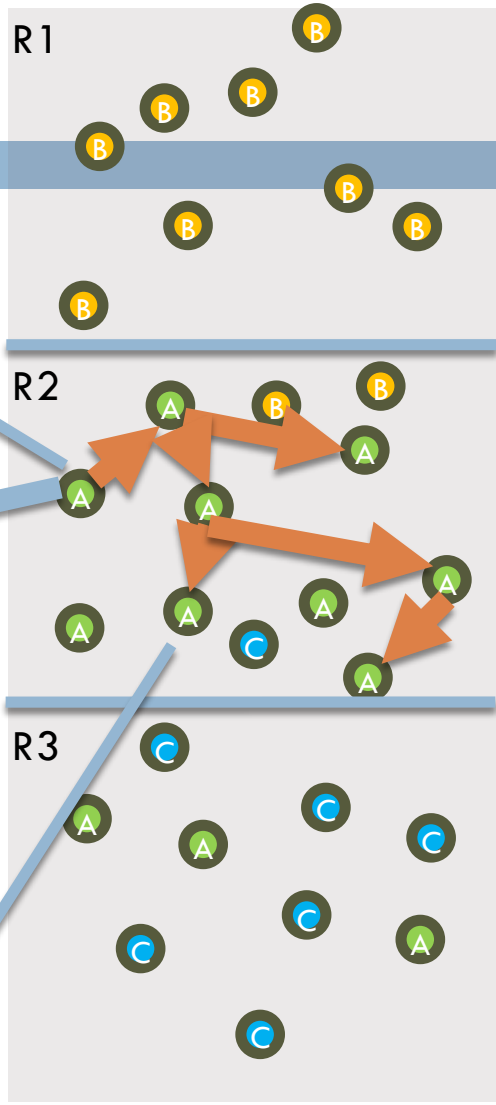
Data Aggregation Phase

Data is encrypted using PIP symmetric keys, aggregated homomorphically and sent to the user

The user decrypts data using PIP aggregate keys

The user can generate aggregate PIP keys from the aggregate session key

Nodes generate the symmetric PIP keys from their individual session keys



Modifying access at run time

- Data generated by a sensor network may hold different degree of importance at different times.

- *Example: A WSN is deployed in a building to keep track of the number of people passing through the building. This information is public and may be accessed by anyone. If however, there is an event in the building, the authorization level of this information is escalated so that only authorized personnel may access the information about the event.*

The network owner should have control over the network's data

Performance Evaluation

- The scheme was implemented on Mica2 sensor motes using TinyOS2.x
- The table shows the execution time and energy consumption of the required operations

Operations	Exec. Time (ms)	Energy Con. (mJ)
Scalar Multiplication	2551	61.224
Ext. Field Multiplication	89	2.136
Ext. Field Cubing	3	0.072
Ext. Field Exponentiation	689	16.536
Paillier Key Generation	453	10.872
Point Addition	116	2.784
Optimised Scalar Mult.	857	20.568

Performance Evaluation

Comparison with other schemes

Scheme	Scalar Mul.	Extension Field Mul.	Extension Field Exp.	Paillier Key-Gen
FDAC	$\|A_j\|+1$	1	1	0
Ruj et. al	$\|A_j\|+1$	1	1	0
Ours	$\|A_j\|+1$	1	1	1

Computation complexity on nodes

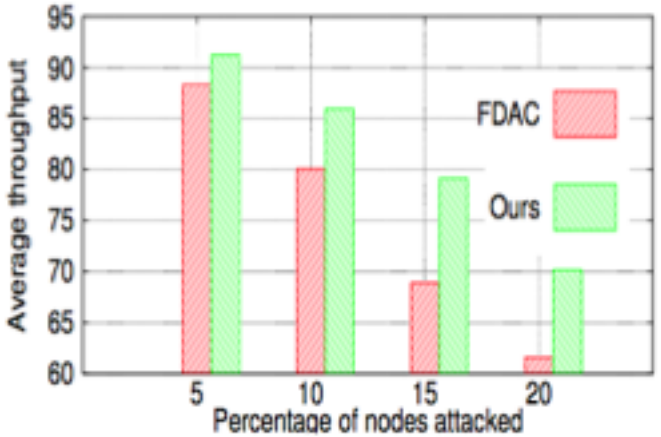
Our scheme requires an additional Paillier key-gen operation, however compared to the computation overhead of the scalar operations, the extra Paillier key-gen operation is negligible

Scheme	Access Control	User Revocation
FDAC	$(\|A_j\|+1)\mathbb{G}_1+1\mathbb{G}_T$	$1\mathbb{G}_T+1\mathbb{G}_1$
Ruj et. al	$(\|A_j\|)\mathbb{G}_1+1\mathbb{G}_2+1\mathbb{G}_T$	$\|A_j\|\mathbb{G}_1$
Ours	$(\|A_j\|+1)\mathbb{G}_1+1\mathbb{G}_T$	$1\|identifier\|$

Communication complexity on nodes

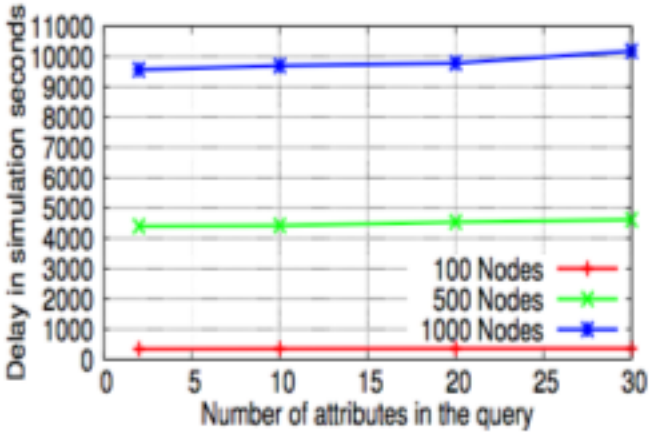
Communication complexity for establishing access control is same for all three schemes but our scheme has a much lower complexity for user revocation

Performance Evaluation



Average throughput with nodes under attack

We assume that nodes under attack do not send any data. Our scheme on account of being distributed performs better under attack



Delay in Query tree formation

--

We simulated three networks in TOSSIM with 100, 500 and 1000 nodes. The figure shows that the number of attributes has a very small impact on the delay in forming the query tree, which is dominated more by the size of network

END OF PRESENTATION
THANK YOU

QUESTIONS