Cloud Based IoT Application Provisioning (The Case of Wireless Sensor Applications)

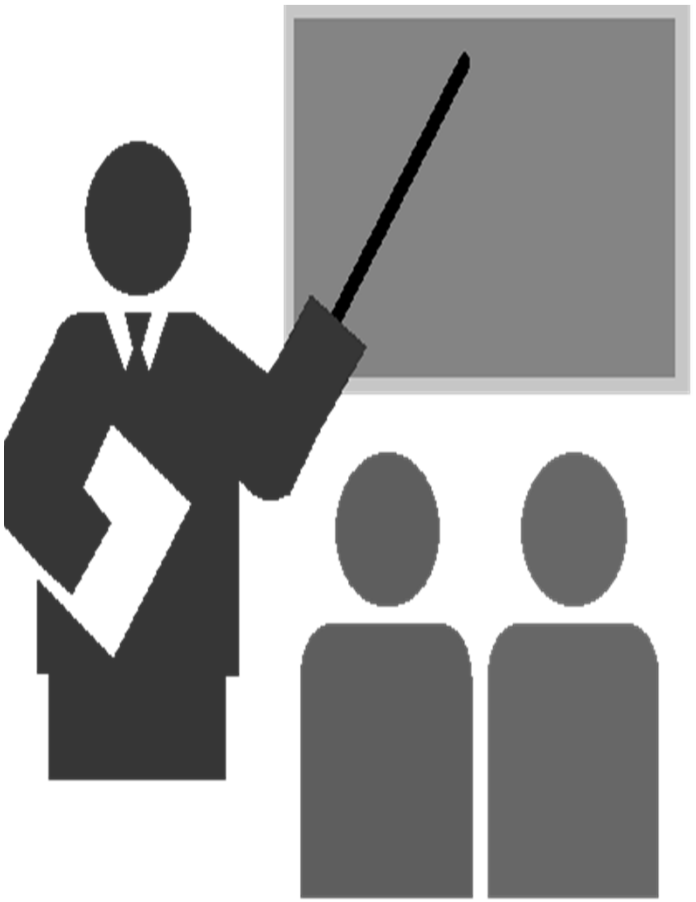
(ENCS 691K)

Roch Glitho, PhD

Associate Professor and Canada Research Chair

My URL - http://users.encs.concordia.ca/~glitho/

Cloud Based IoT Applications Provisioning: The Case of Wireless Sensor Networks



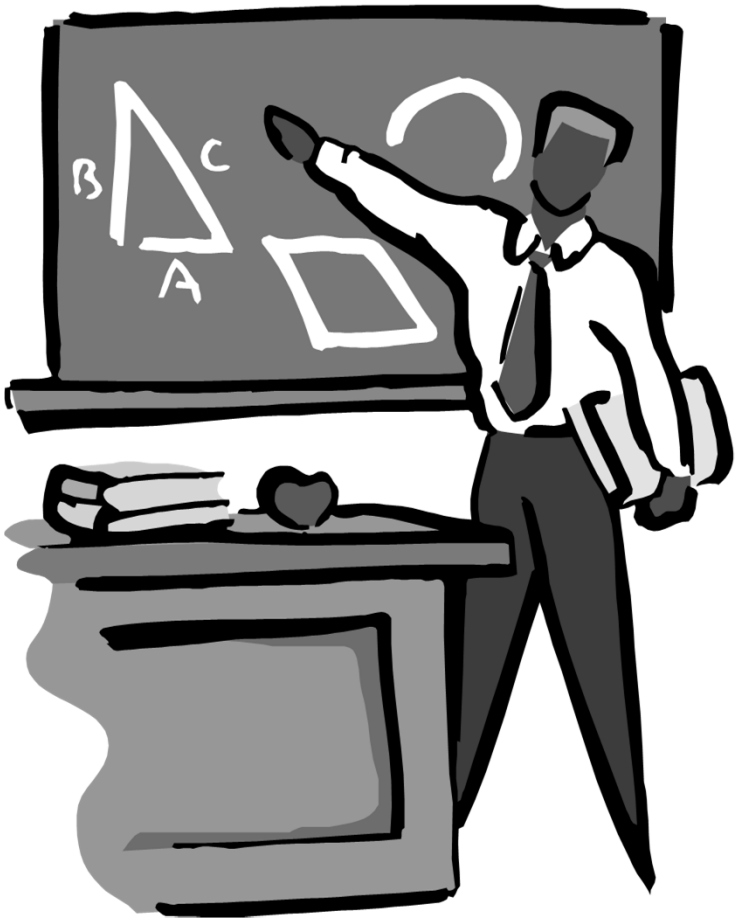
- On Wireless Sensor Networks
- Cloud Based WSN applications provisioning

Internet of Things

“Things such as RFID tags, sensors, actuators, mobile phones which are able to interact with each other and cooperate with their neighbours to reach common goals”

L. Atzori et al, The Internet of Things: A Survey, Computer Networks (54), 2010

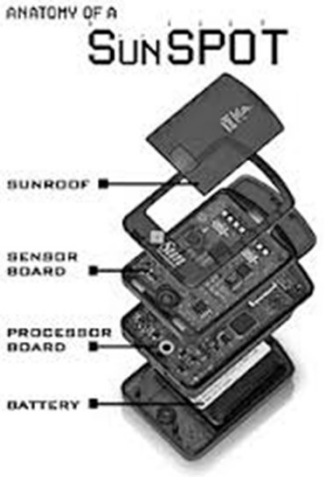
On Wireless Sensor Networks



- **Basics**

Basics

Wireless Sensors



Wireless Sensors

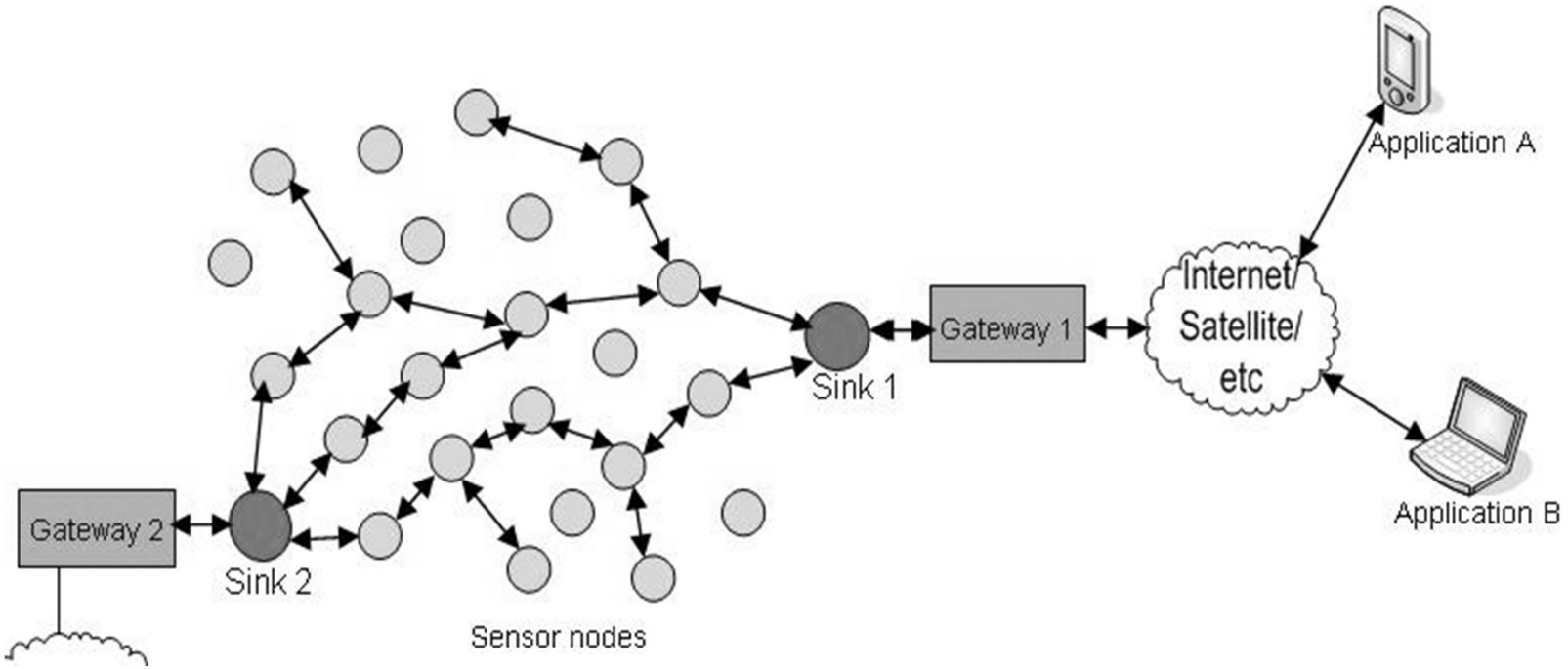
Small scale autonomous devices that can sense, compute and communicate ambient information

- Ambient information
  - Space
    - e.g. location, velocity
  - Environment
    - e.g. luminosity, level of noise
  - Physiology
    - E.g. blood pressure, heartbeat

Conventional Wireless Sensor Networks (WSNs)

- Sensors
  - Do the actual sensing
- Aggregators
  - Logical representatives of regions of interest
    - Summarize data for regions
- Sinks
  - Collect data from all sensors / aggregators
    - Interact with end – user services / applications via gateways
- Gateways
  - Dual interfaces
    - Bridge WSNs and outside world

Conventional Wireless Sensor Networks

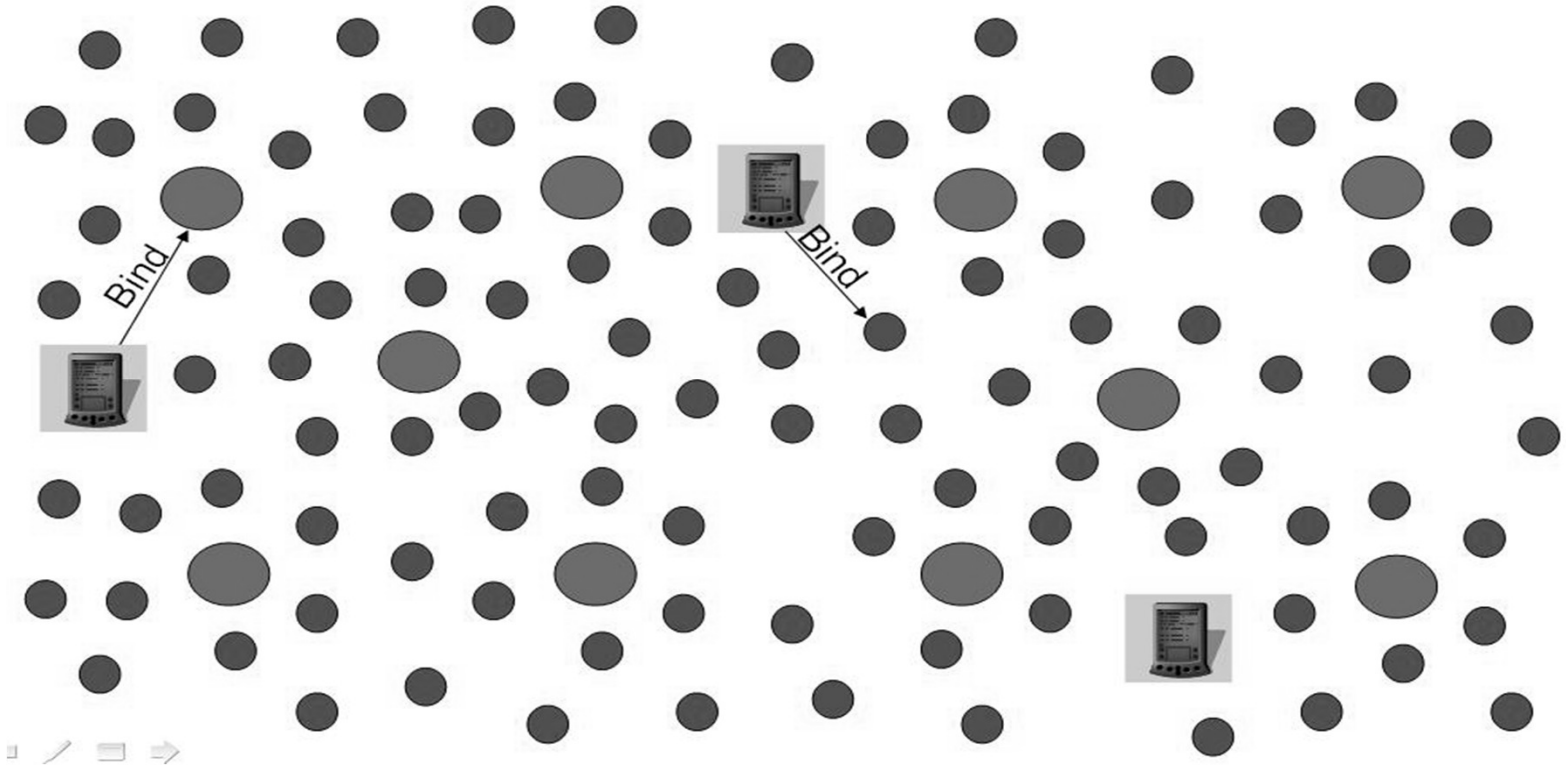


Sink-Less Wireless Sensors Networks

No sink, No gateway

- End-user services / applications interact directly with individual sensors
- Use cases
  - Battlefield assessment
    - Sensors scattered over a field to detect landmine
    - Soldiers moving in the field with application devices
  - Rescue operations
    - Indoor monitoring
      - Fire fighters

Sink-Less Wireless Sensor Networks

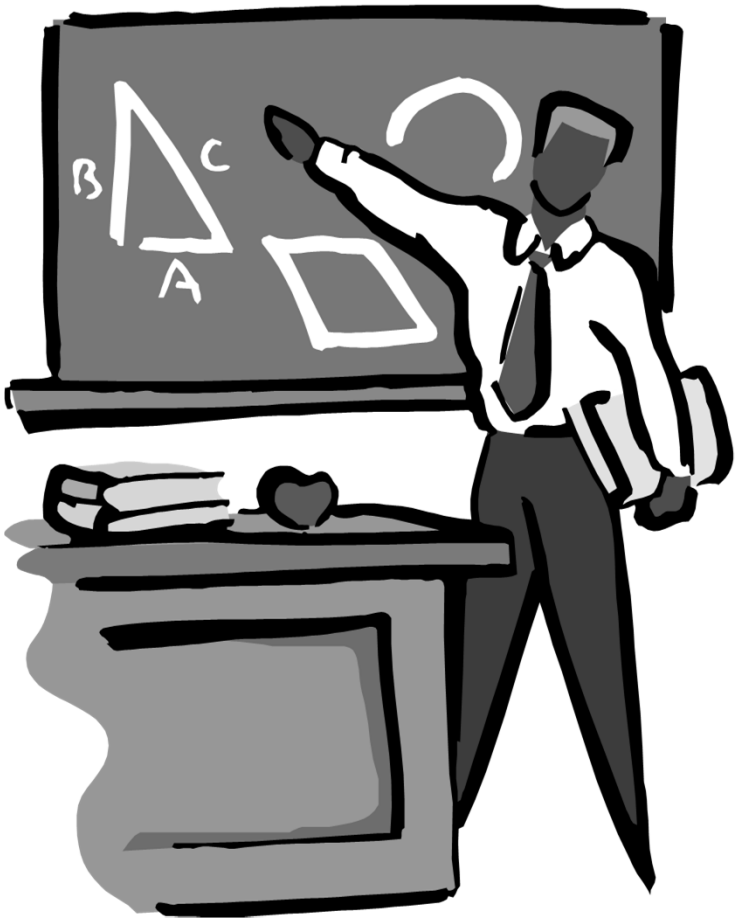


Applications areas

Numerous

- Military
- Environment
- Health
- Home
- Industry

Cloud Based WSN Applications Provisioning



- **Use of cloud storage and processing power**
- **Applying cloud fundamentals to WSN Application Provisioning (WSN Virtualization)**

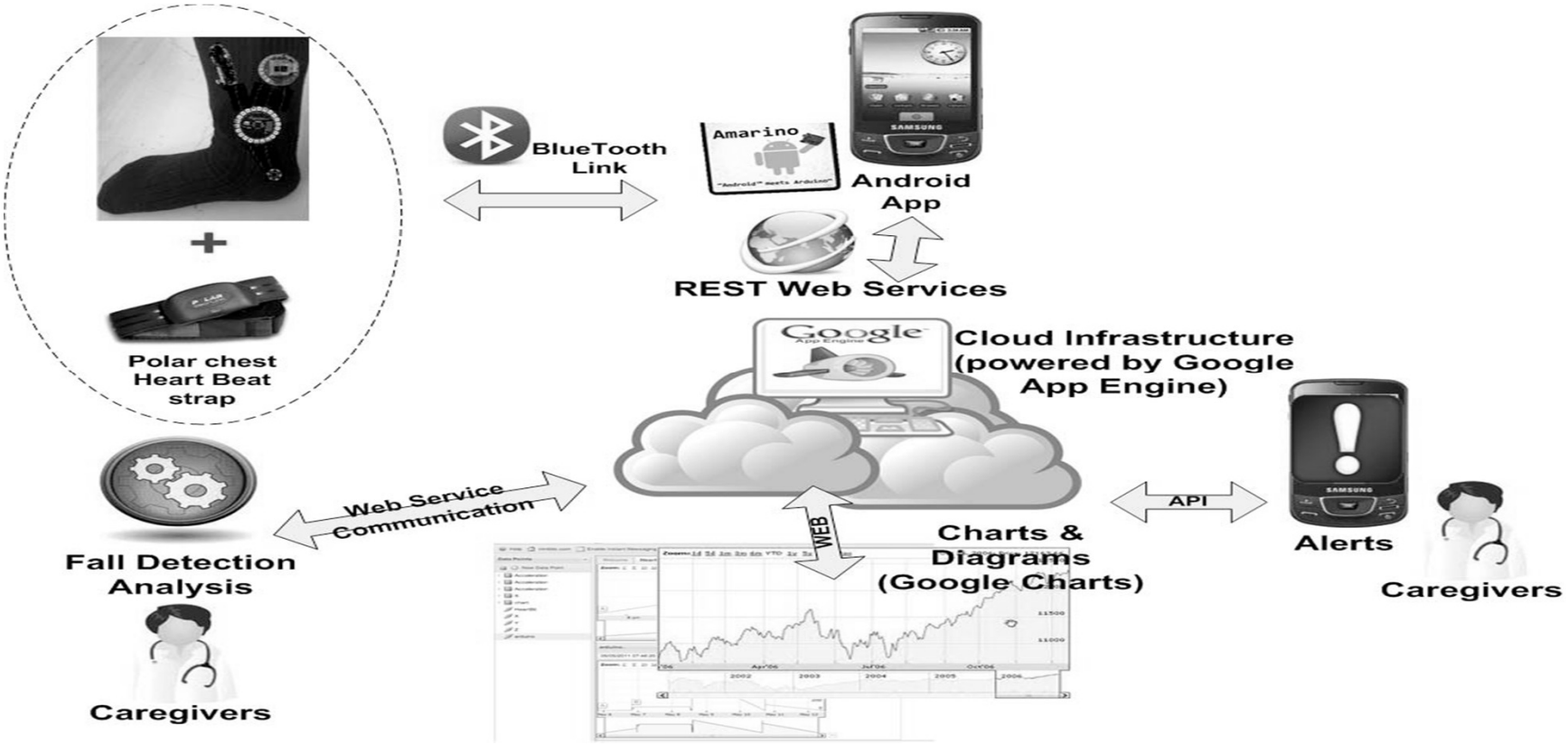
Use of Cloud Processing and Storage Power

Use of Cloud Processing and Storage Power

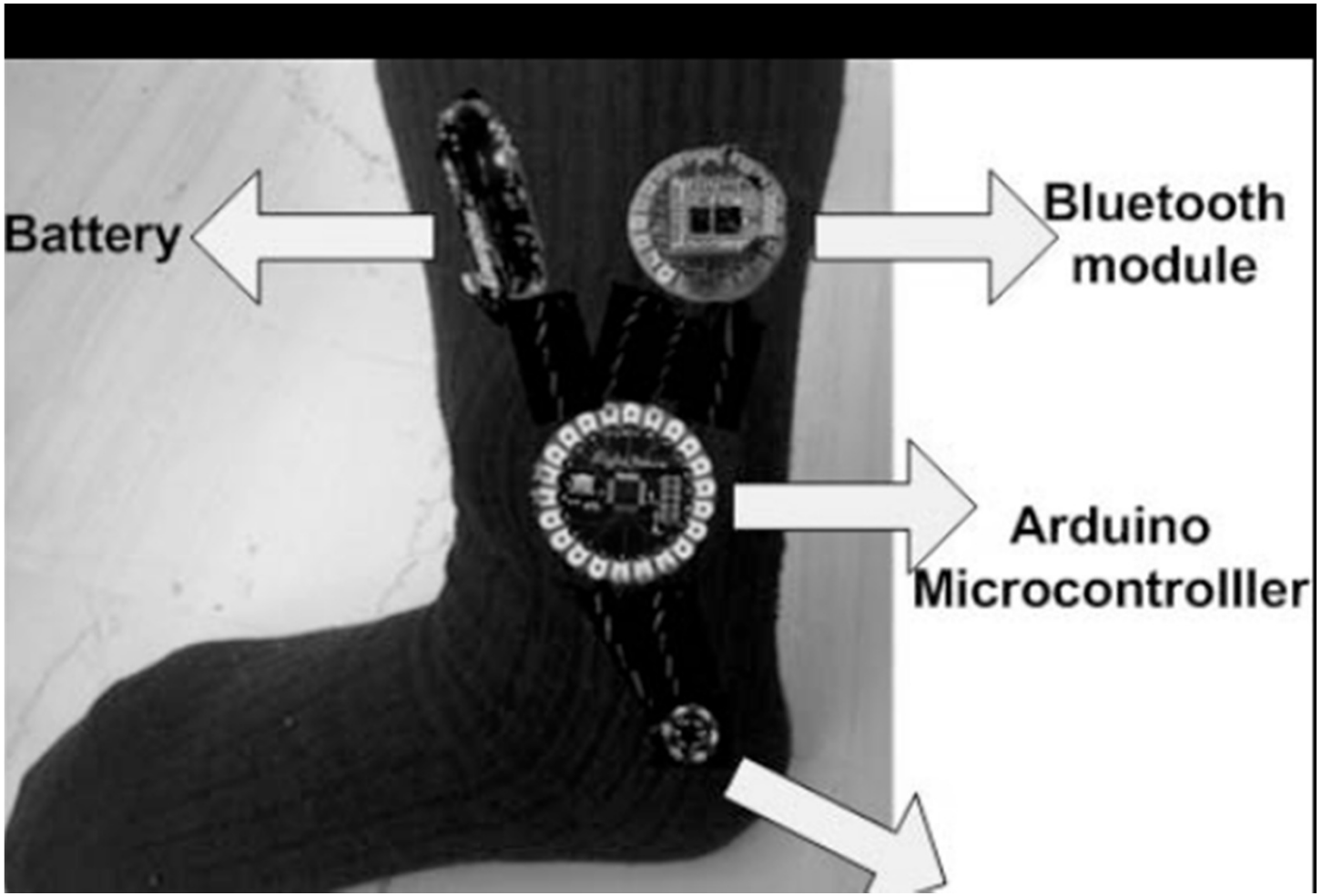
An illustration:

C. Doukas and I. Maglogiannis, Managing Wearable Sensor Data through Cloud Computing, 2011 Third International Conference on Cloud Computing Technology and Science

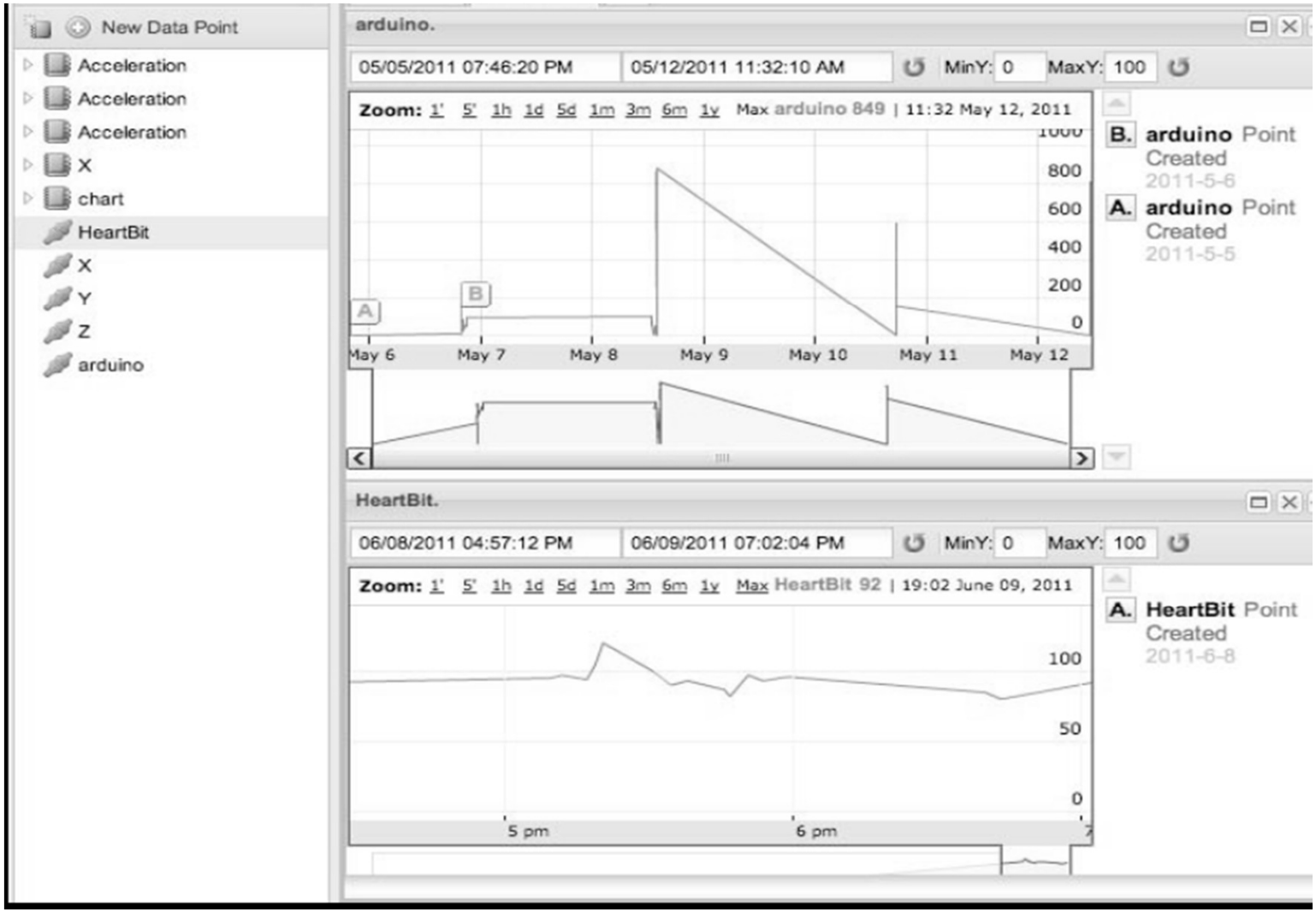
Managing Wearable Data Through Cloud



Managing Wearable Data Through Cloud



Managing Wearable Data Through Cloud



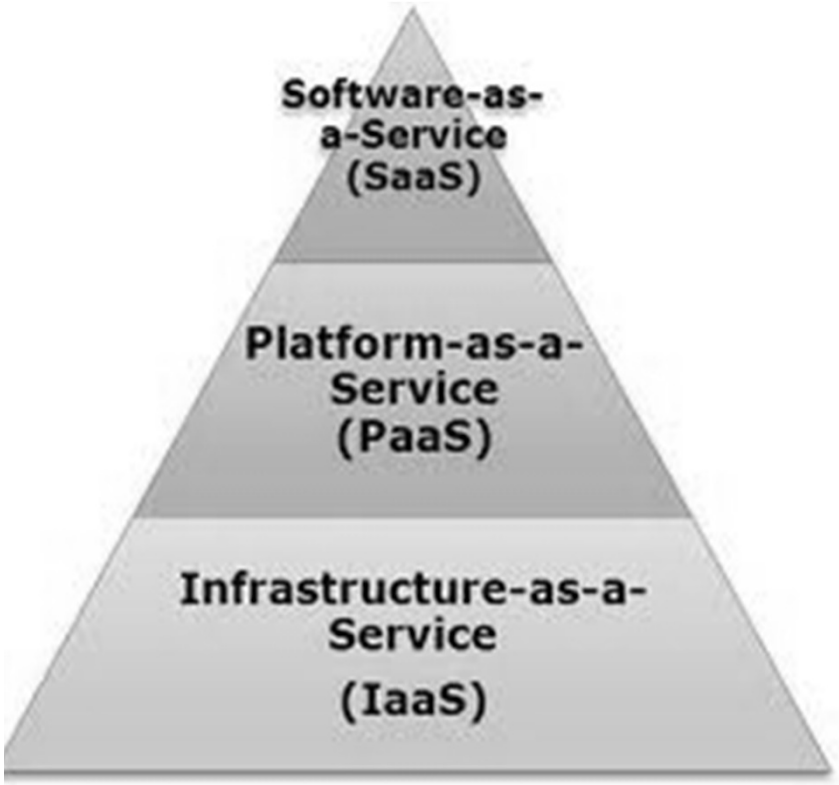
Use of Cloud Storage and Processing

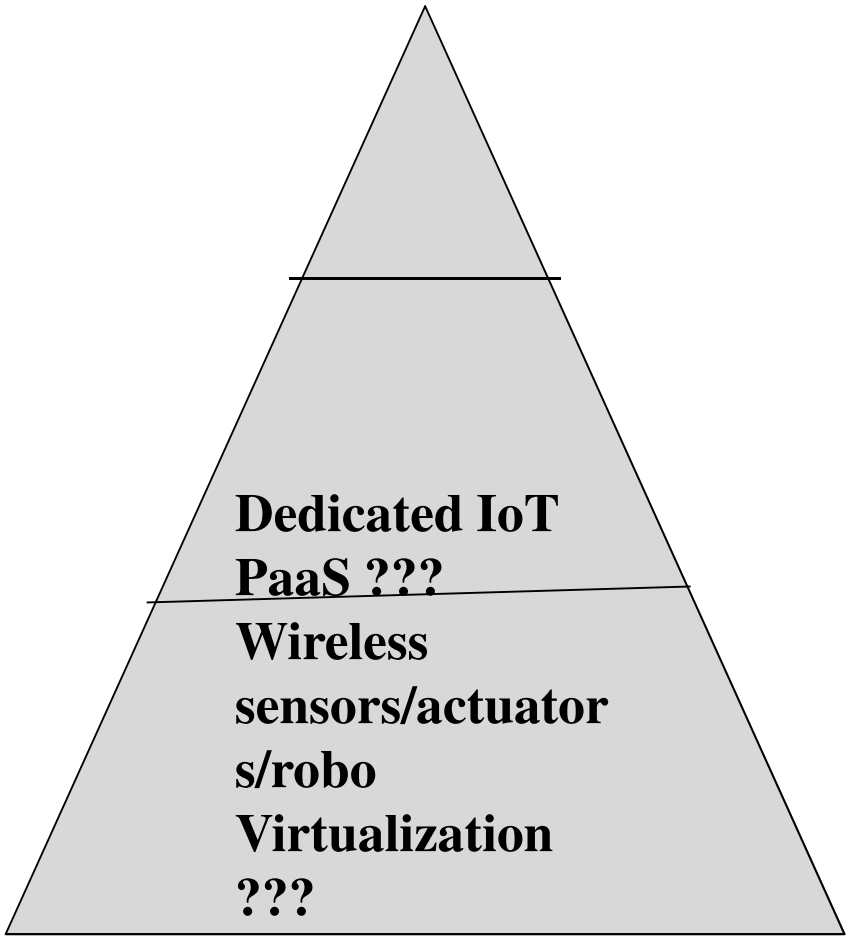
What is used from cloud computing:

- IaaS
  - Resource for storing and processing data sent by sensors
- PaaS
  - General platform for applications provisioning

Use of Cloud Storage and Processing

What more could have been used to reap all the benefits (e.g. efficiency in resource usage, easy application provisioning)





Applying Cloud Fundamentals to WSN Apps
Provisioning: WSN Virtualization

Virtual sensors (VS) instead of virtual machines (VM)?

References

I Khan, F. Belqasmi, R. Glitho, N. Crespi, M. Morrow, P. Polakos, Wireless Sensor Network Virtualization: A Survey, IEEE Communications Surveys and Tutorials , Vol PP, Issue 99, March 2015

- I Khan, F. Belqasmi, R. Glitho, N. Crespi, M. Morrow, P. Polakos, Wireless Sensor Network Virtualization: Early Architecture and Research Perspectives, IEEE Network , May / June 2015

WSN Virtualization: Motivations

Current situation

- Applications bundled with WSN at deployment time
- No possibility to re-use the deployed WSN for other applications
- Deployment of redundant WSNs

WSN Virtualization: Motivations

What could WSN virtualization bring?

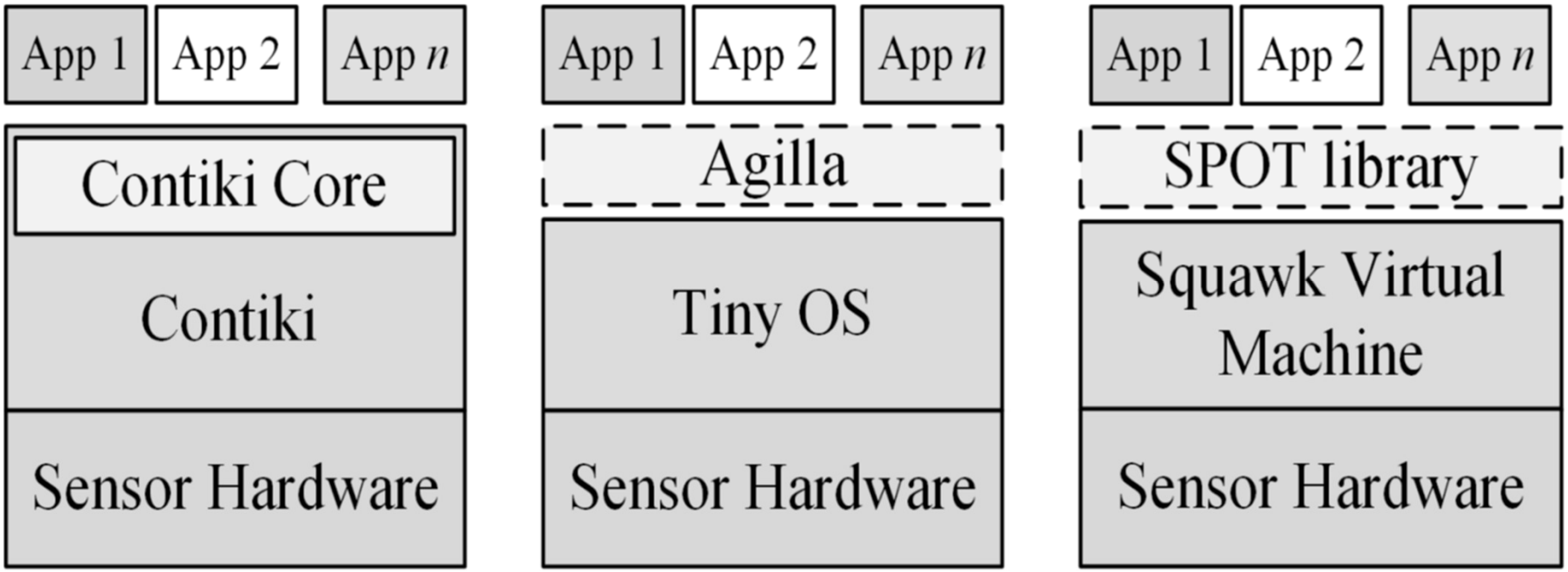
- Efficiency in resource usage through the sharing of a same WSN infrastructure by several different applications with the possibility of deploying new applications after the deployment of the WSN infrastructure

WSN Virtualization: Motivations

A scenario

- Citizens might own a sensor in their houses to detect fire
- If the city administration decides to deploy an application which shows the fire contour (e.g. fire direction, intensity), there are 2 approaches:
  - Redeploy sensors everywhere including citizen houses
  - Deploy sensors in streets / parks and re-use the sensors already deployed in citizen houses
    - Sensors running in citizen houses will then run two different tasks
      - The task allocated by house owner (e.g. detect fire)
      - The task allocated by the city administration (fire contour algorithm)

WSN Virtualization: Node Level Virtualization

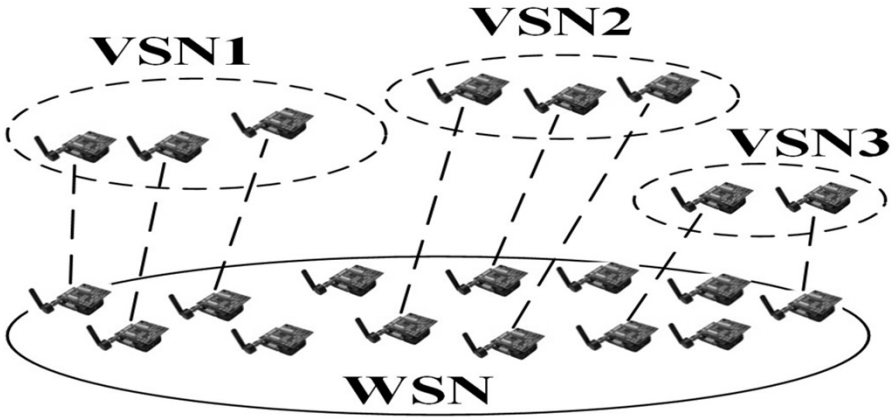


a) OS-based solution (e.g. Contiki)

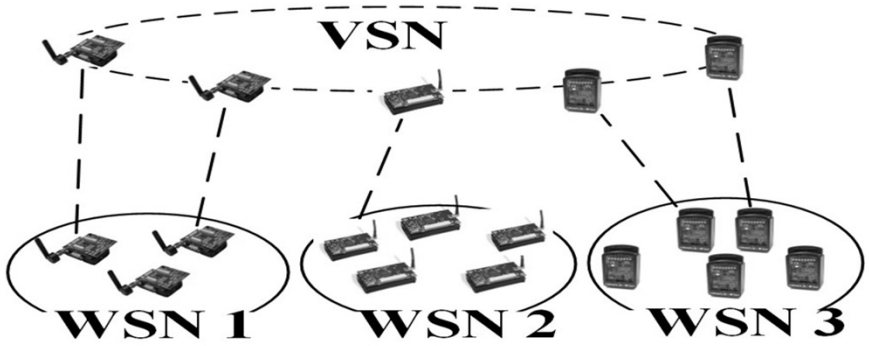
b) Middleware-based solution (e.g. Agilla)

c) Virtual machine-based solution (e.g. Squawk VM)

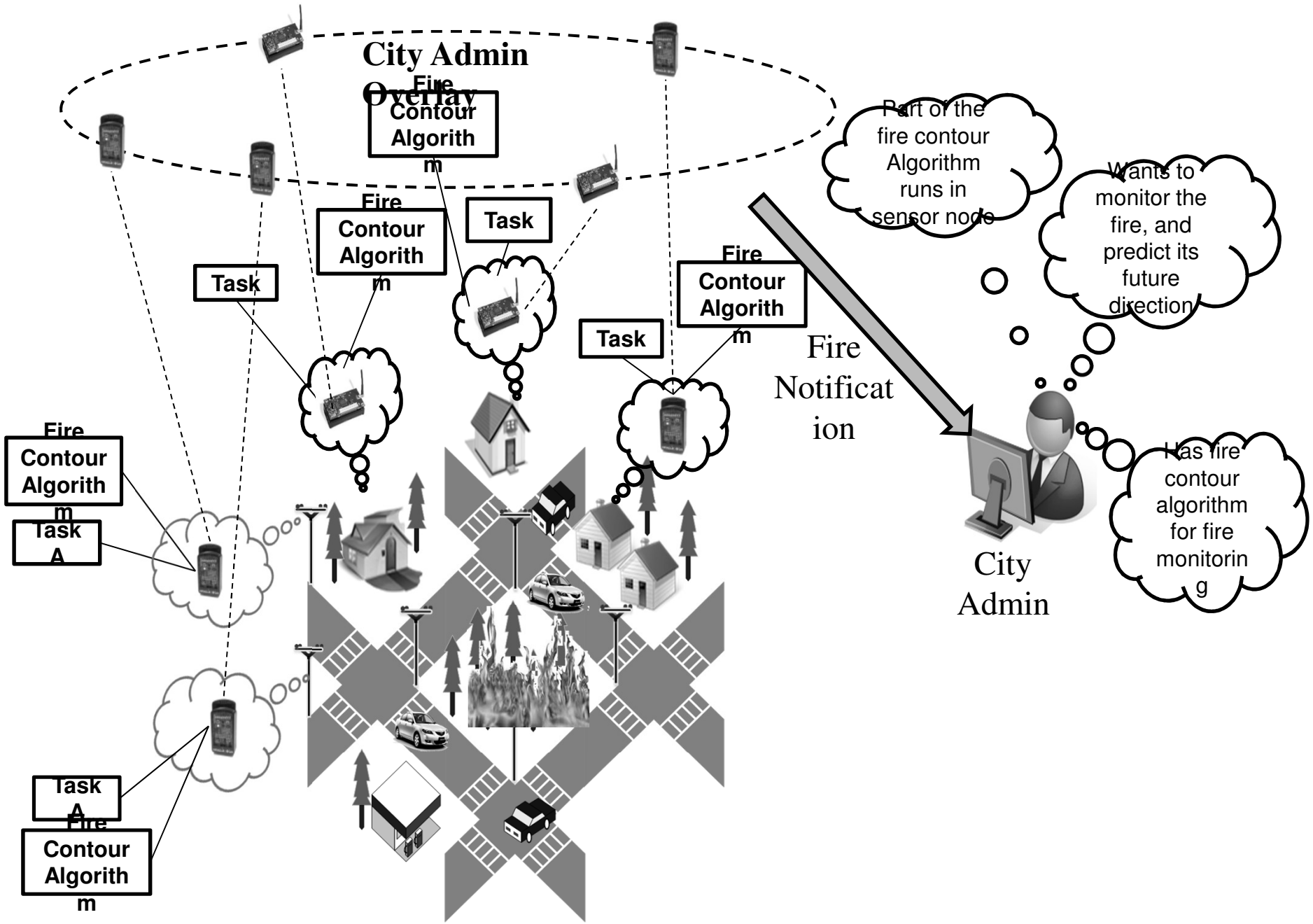
WSN Virtualization: Network Level Virtualization



a) Multiple VSNs over single WSN

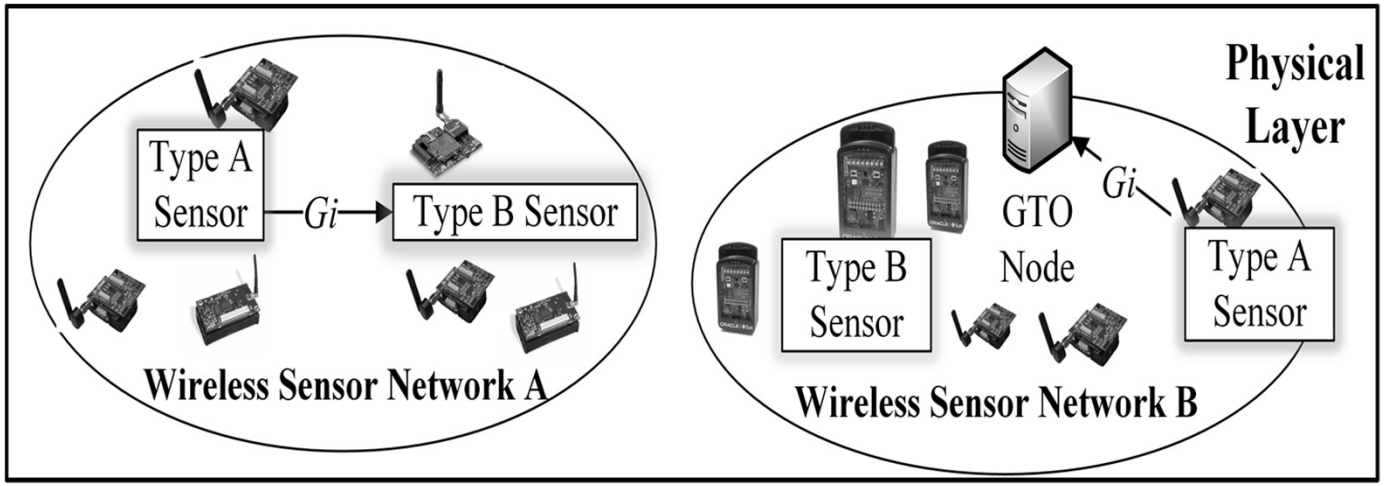


b) Single VSN over multiple WSNs


City Admin Overlay
Fire Contour Algorithm
Task
Fire Contour Algorithm
Task
Fire Contour Algorithm
Task
Fire Contour Algorithm
Task A
Task A
Fire Contour Algorithm
Part of the fire contour Algorithm runs in sensor node
Wants to monitor the fire, and predict its future direction
Fire Notification
City Admin
Has fire contour algorithm for fire monitoring

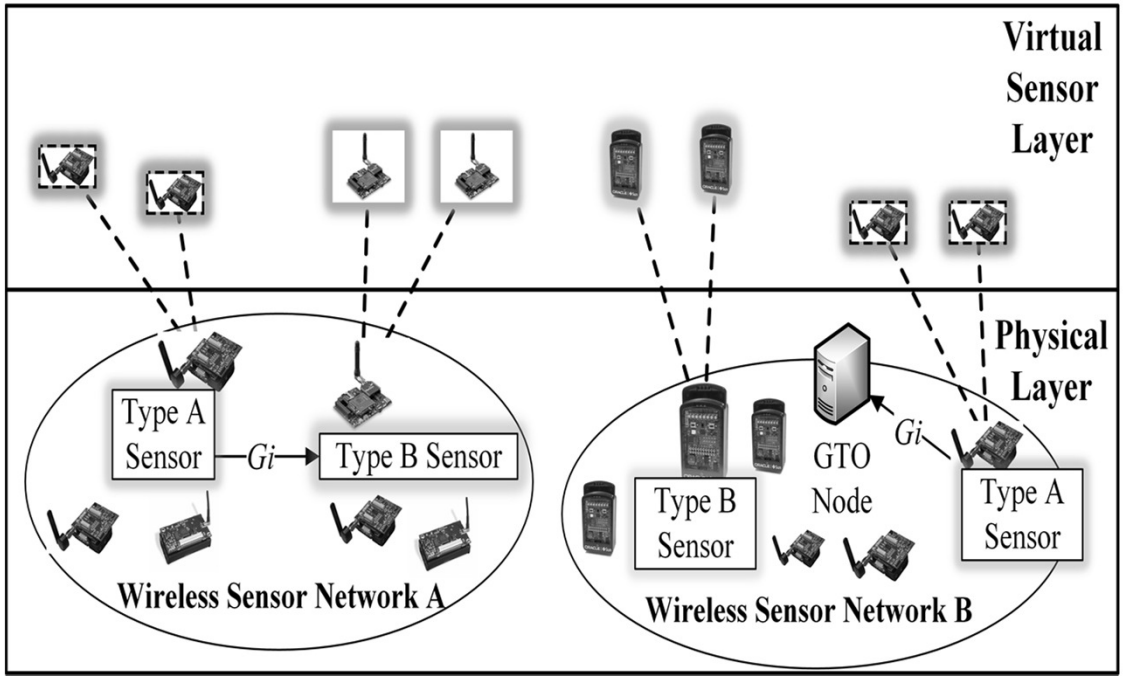
WSN Virtualization Architecture

- Physical Layer
  - Two types of sensors
    - Resource constrained (Type A)
    - Capable sensors (Type B)
  - Gates-to-Overlay (GTO) nodes



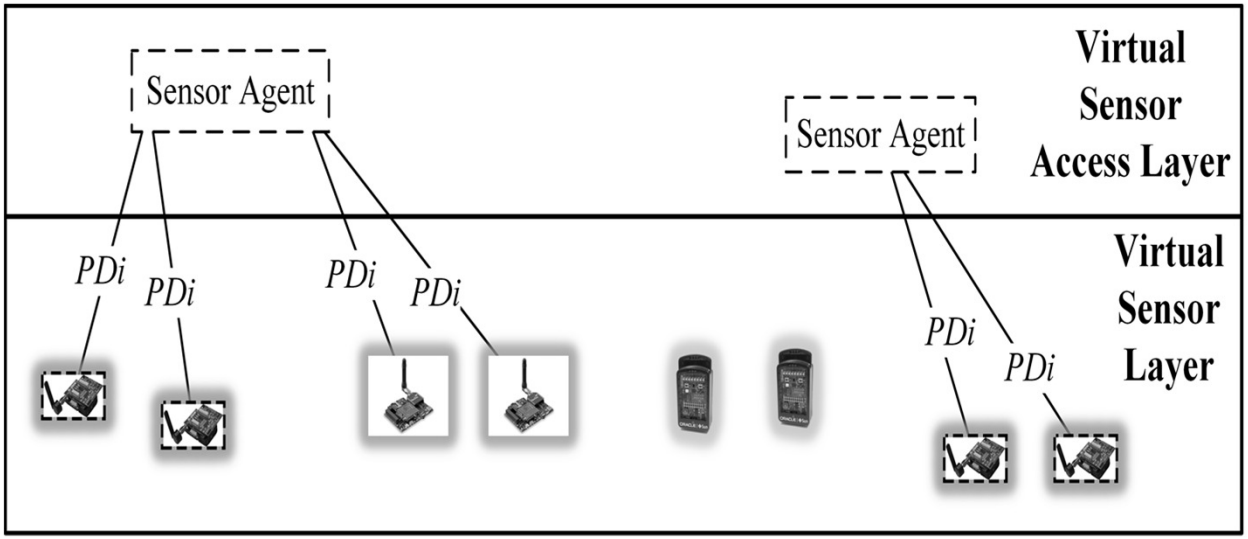
WSN Virtualization Architecture

- Virtual Sensor Layer
  - Abstracts the multiple tasks run by physical sensors as virtual sensors
    - Each virtual sensor executes a different task for an application



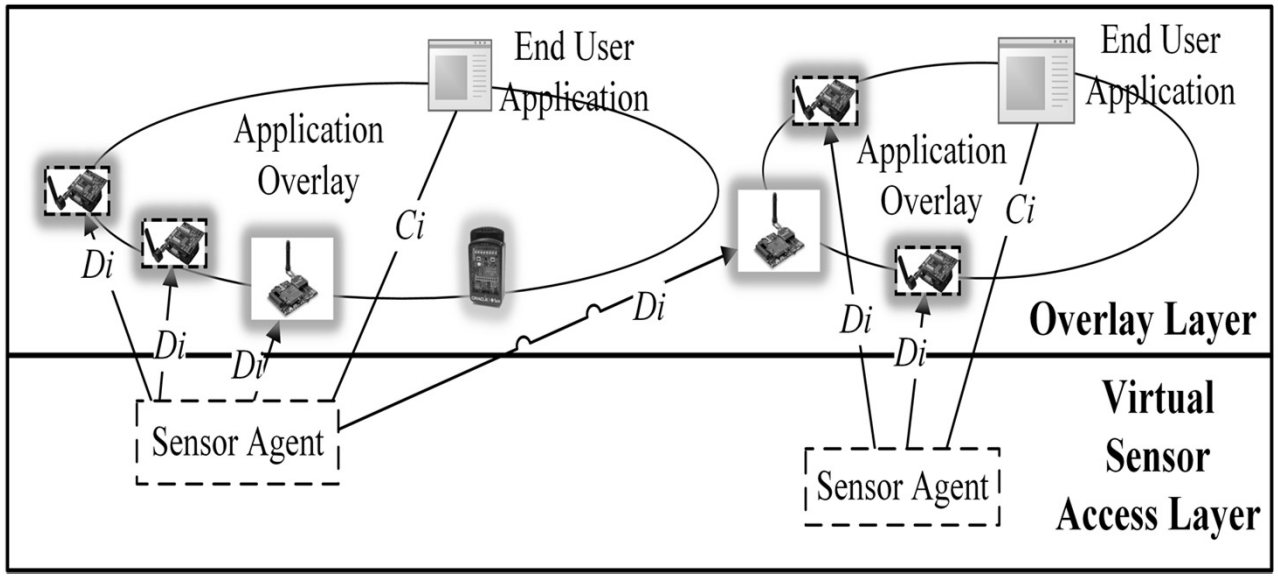
WSN Virtualization Architecture

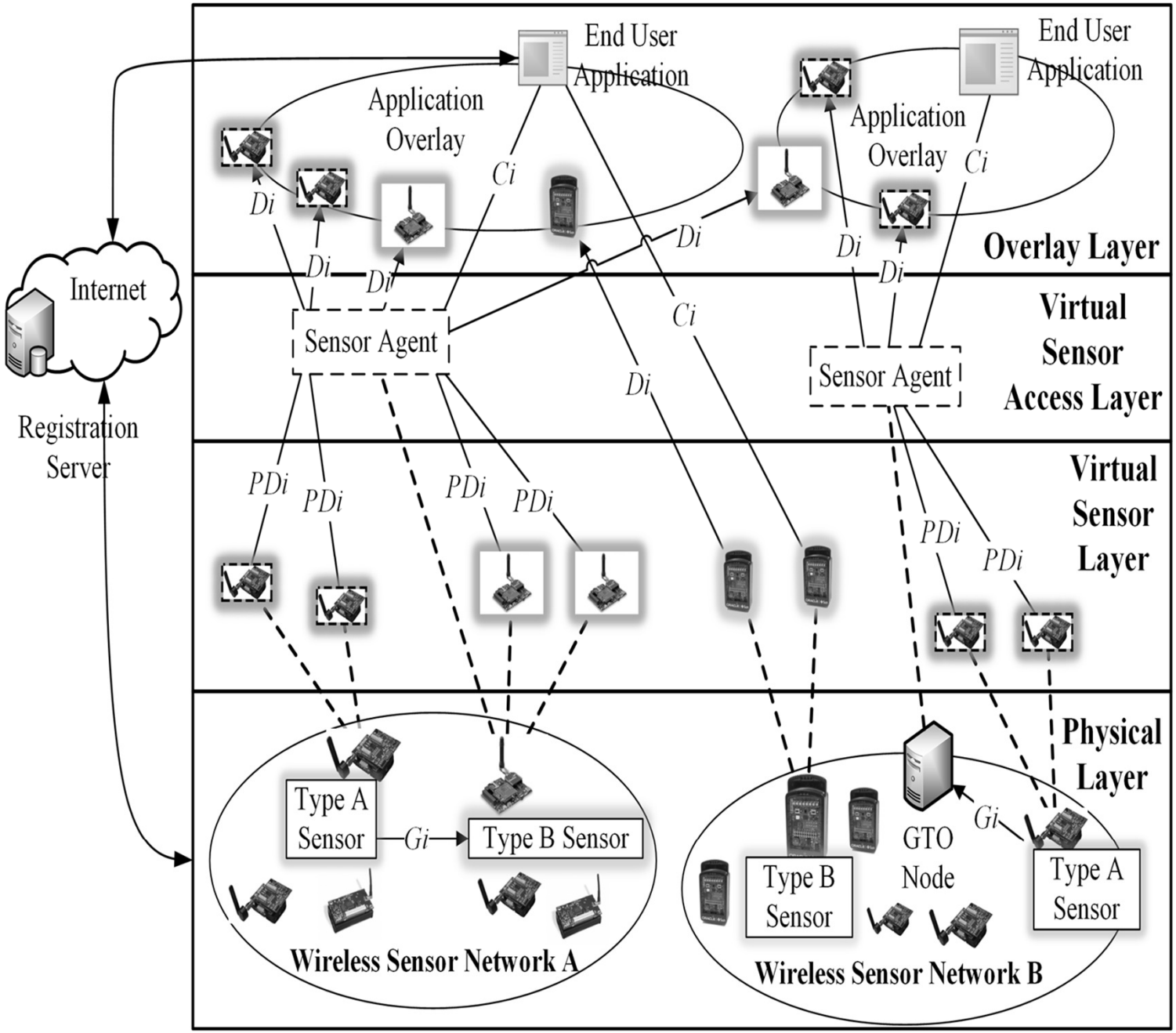
- Virtual Sensor Access Layer
  - Consists of Sensor Agents (e.g. gateways) to provide supplier/sensor brand independence
    - Abstracts virtual sensors and interacts with the applications/services



WSN Virtualization Architecture

- Application Overlay Layer
  - Consists of independent application overlays
    - Interact with Sensor Agents using a standardized interface (e.g. SenML)




End User Application
End User Application
Application Overlay
Application Overlay
Ci
Ci
Di
Di
Di
Di
Di
Di
Di
Di
Overlay Layer
Internet
Registration Server
Sensor Agent
Sensor Agent
Virtual Sensor Access Layer
Ci
Di
PDi
PDi
PDi
PDi
PDi
PDi
Virtual Sensor Layer
Physical Layer
Type A Sensor
Gi
Type B Sensor
Wireless Sensor Network A
Type B Sensor
GTO Node
Gi
Type A Sensor
Wireless Sensor Network B

The End