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

(ENCS 691K – Chapter 7)

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

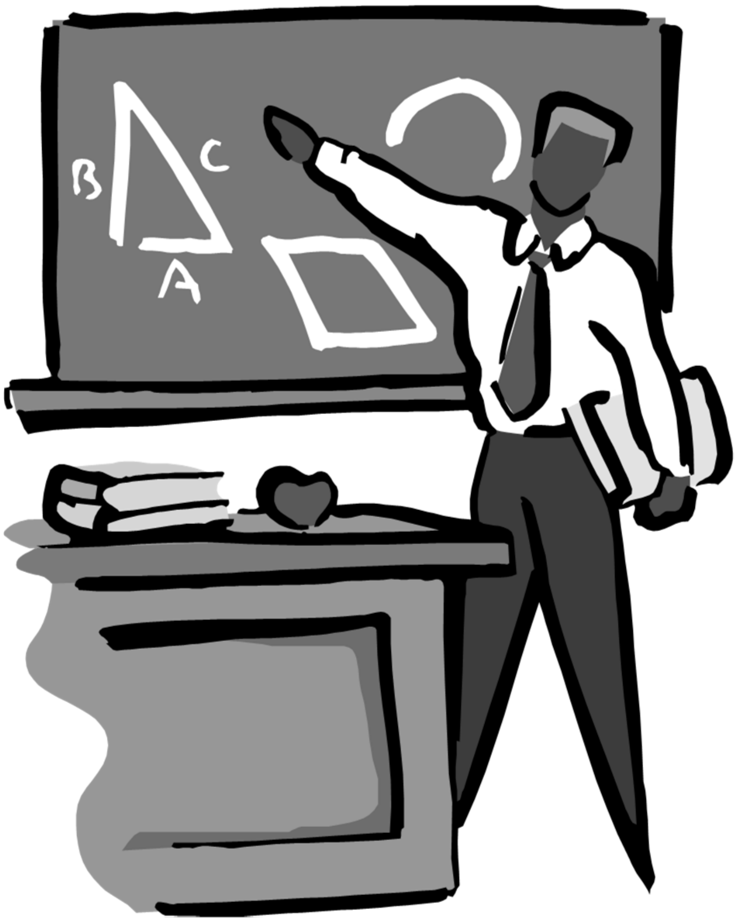
- On Wireless Sensor Networks
- Cloud Based WSN applications provisioning

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



- Introduction
- IPv6 Enabled Wireless Sensors
- RESTful Web Services for WSN: Constrained Application Protocol

Introduction

Wireless Sensors

Wireless Sensors

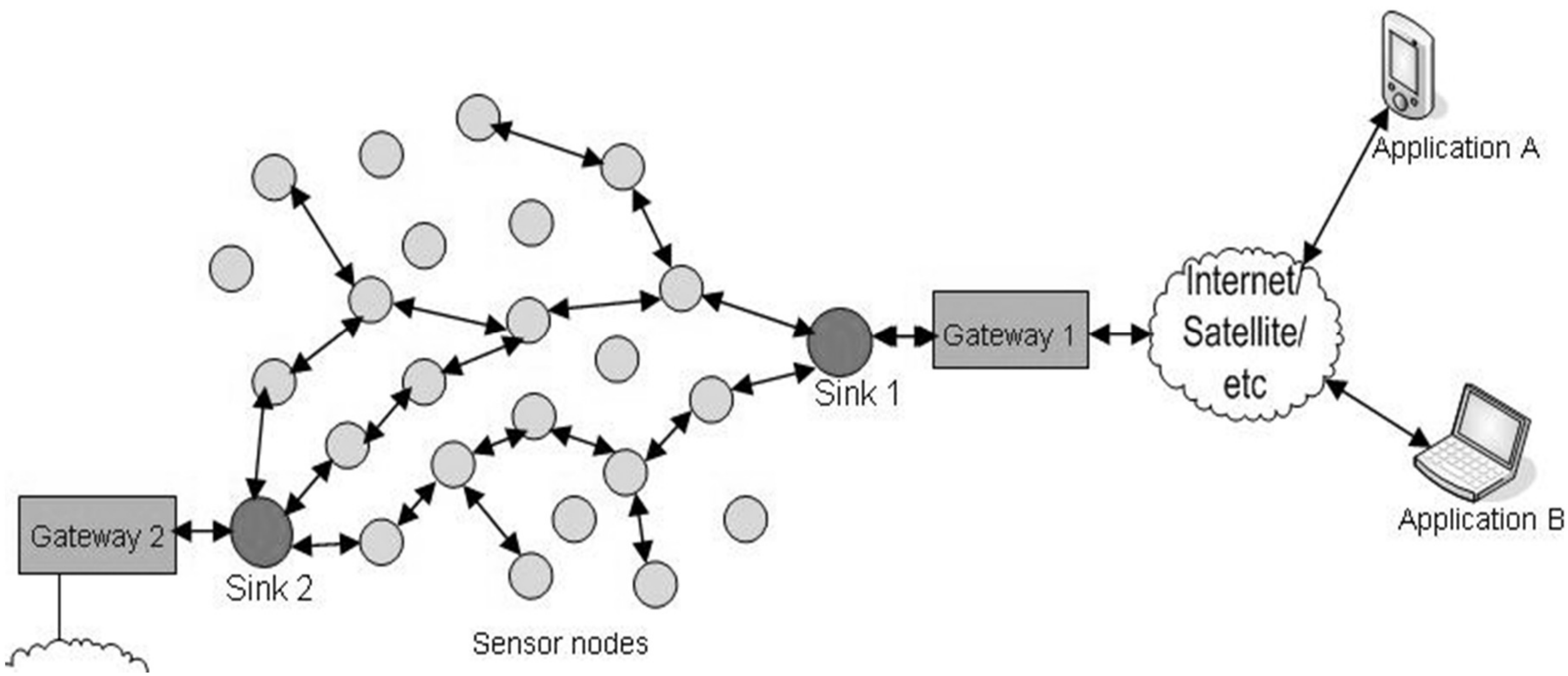
Small scale autonomous devices that can sense, compute and communicate ambient information

- 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

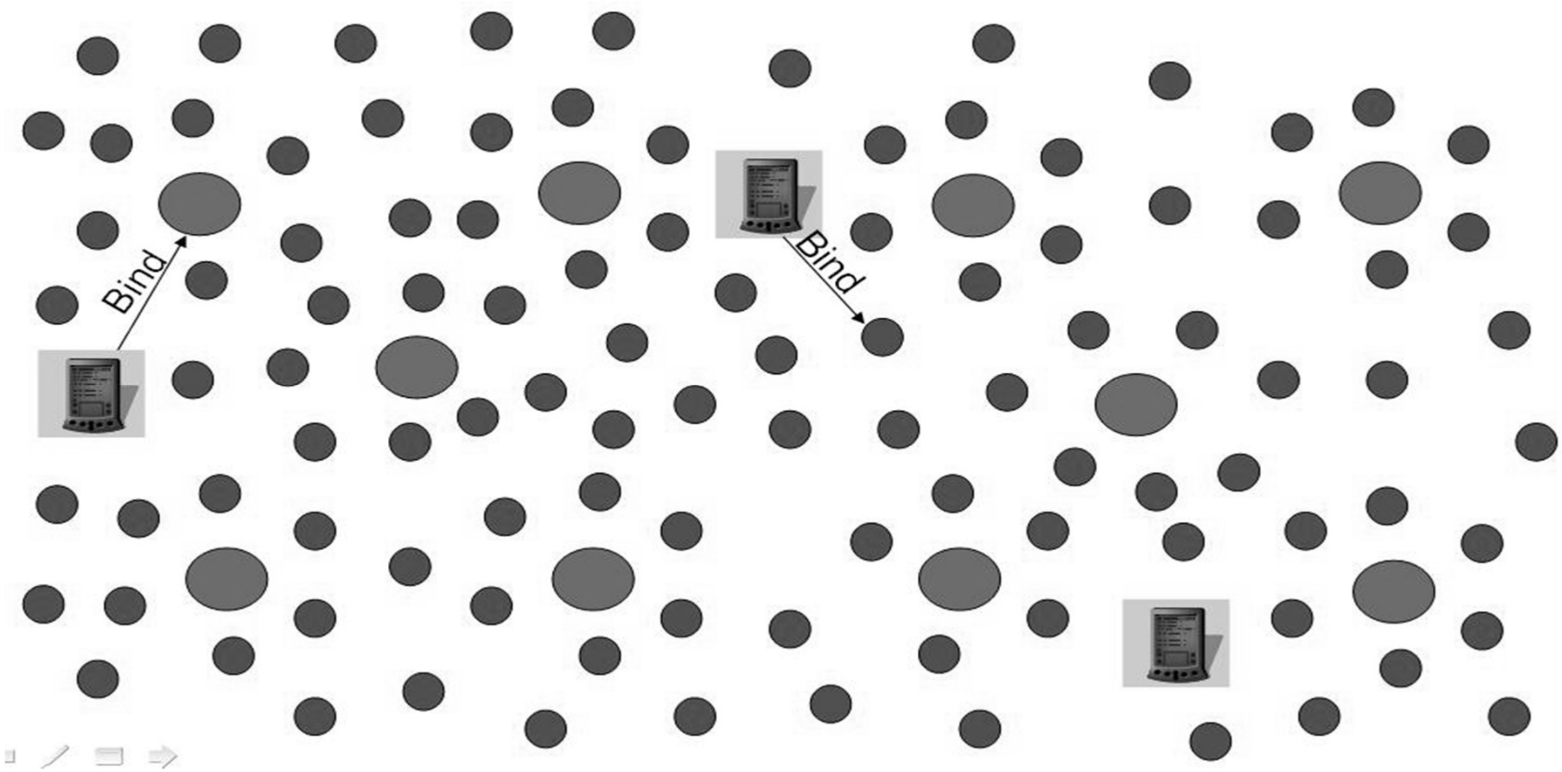


Sink-Less Wireless Sensors Networks

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



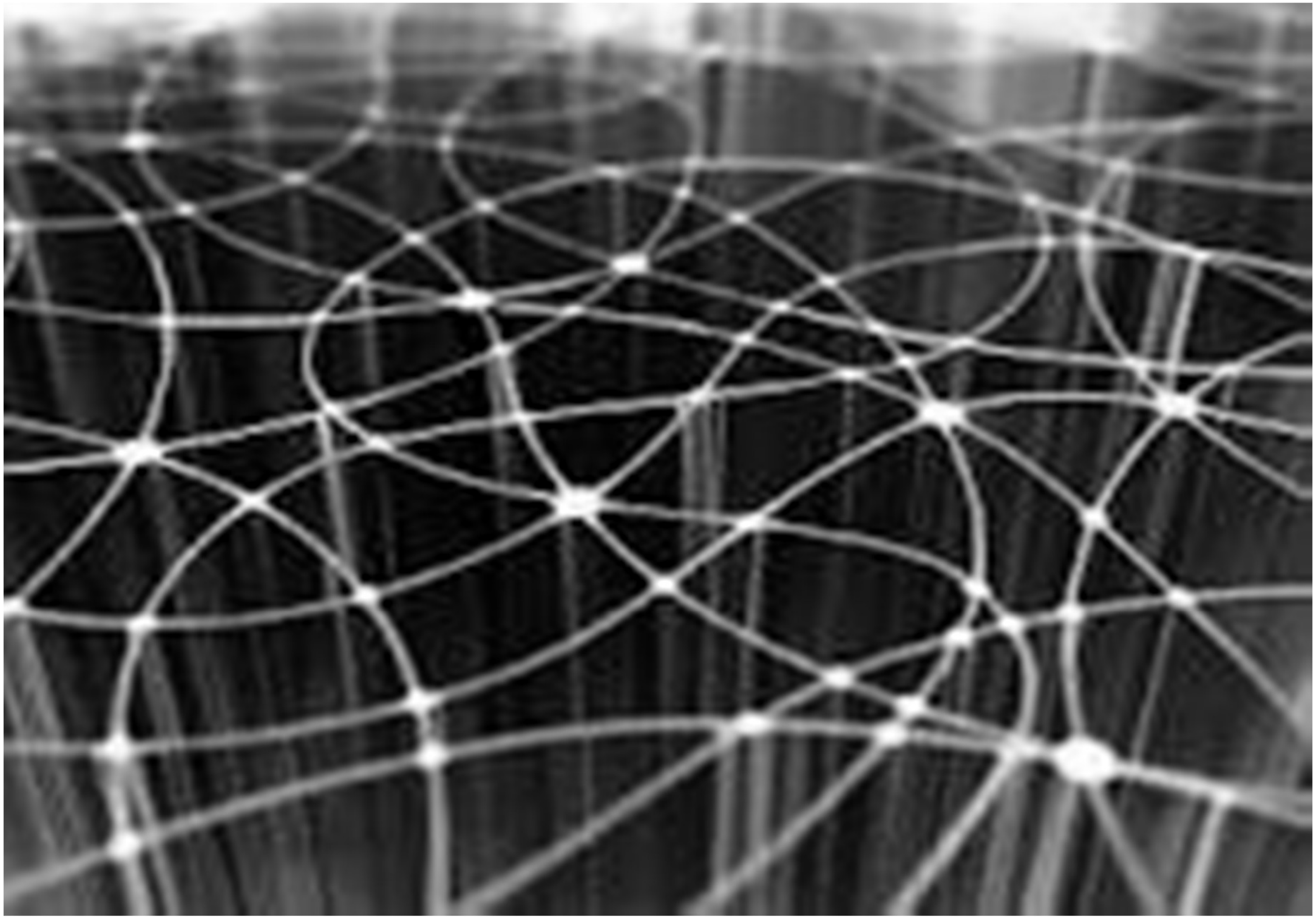
Applications areas

Numerous

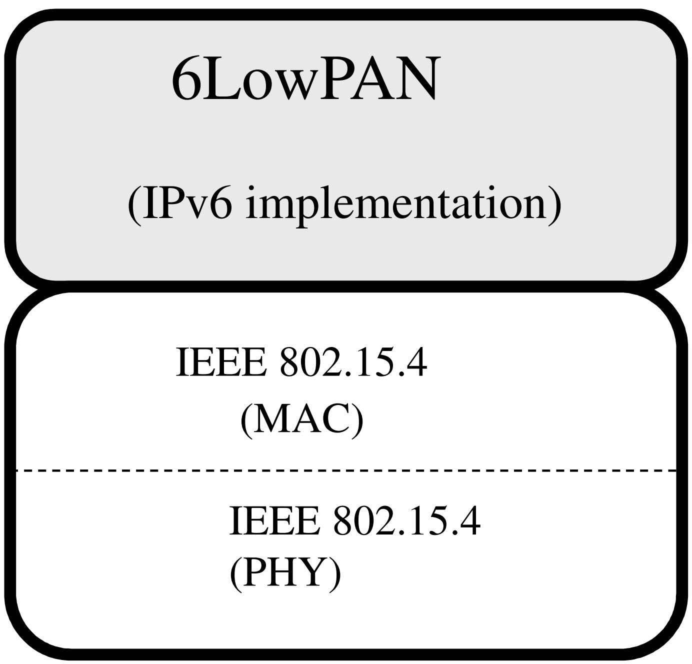
- Military
- Environment
- Health
- Home
- Industry

IPv6 Enabled Wireless Sensors

Communications and networking



Building on IEEE 802.15.4: 6LoWPAN



Networking level supported by more
And more sensors

**Standard Communications
Level (MAC + PHY)
supported by more and more sensors**

6LoWPAN

- IETF Worked on how to layer IPv6 on low data, low power and small footprint radio networks such as IEEE 802.15.4
  - Motivations
    - “Easy” re-use the wealth of transport and application protocols that exist in the IP world
    - Easy interoperability with the IP world
    - Re-use of the large IPv6 address space

6LoWPAN

- IETF Work on how to layer IPv6 on low data, low power and small footprint radio networks such as IEEE 802.15.4
  - Key challenge
    - IPv6 Implementation cost
      - 40 bytes IPv6 header transmission
      - IPv6 logic

6LoWPAN

- IETF Work on how to layer IPv6 on low data, low power and small footprint radio networks such as IEEE 802.15.4
  - Technical solution
    - Stateless header compression that allows IPv6 packet transmission in as few as 4 bytes
      - Pay for what you use
        - 4 basic headers that are not all always transmitted

IEEE 802.15.4 PHY

- Direct Sequence Spread Spectrum (DSSS) Access mode
  - Three bands
    - 2450 MHz
    - 915 MHz
    - 868 MHz

IEEE 802.15.4 MAC

- Two types of nodes
  - Full Function Devices
    - Equipped with a full set of MAC layer functions
    - Act as network coordinator and/or end-device
  - Reduced Function Devices
    - End devices only
      - Equipped with sensors/actuators
      - Interact with a single full function device

IEEE 802.15.4 MAC

- Two topologies
  - Star (master slave)
  - Peer to peer
- Protocol
  - CSMA - CA

RESTFul Web Services for Wireless Sensor Networks

RESTFul Web services for M2M: Constrained Environments

Z. Shelby, Embedded Web Services, IEEE Wireless Communications, December 2010

IETF Constrained Application Protocol (CoAP)

- **Constrained Application Protocol (CoAP)**:
  - Realizes a minimal subset of REST along with resource discovery, subscription/notification, and the use of appropriate security measures

CoAP features

- **Compact Header**: binary encoded header (4 bytes) + extensible options , and total header 10-20 bytes for typical requests
- **Methods and URIs**: like HTTP (GET, PUT, POST, DELETE)
- Content types: Can indicate content type of the payload in the header

CoAP features

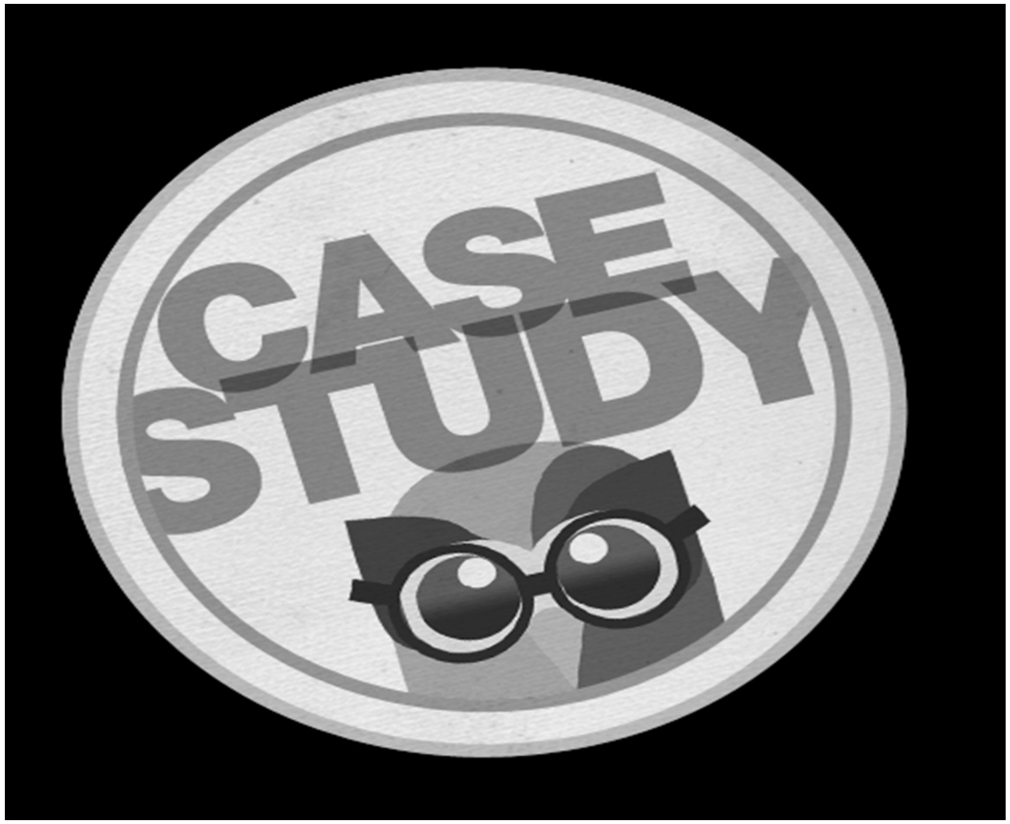
- **Transport binding**: UDP + simple stop-and-wait reliability mechanism. Optional security is supported using Datagram Transport Layer Security (DTLS)

CoAP features

- **Resource Discovery**: to discover the list of resources offered by a device, or for a device to advertise or post its resources to a directory service.

- **Subscription**: an asynchronous approach to support the push of information from servers to clients using subscriptions

Case Study : Integrating Wireless Sensor Networks with the Web



W. Colitti et al,
http://hinrg.cs.jhu.edu/joomla/images/stories/IPSN_2011_koliti.pdf

Design and Development of an End to End Architecture

- CoAP over 6LoWPAN
- Contiki based WSN
- Access of WSN data directly from a browser

Design and Development of an End to End Architecture

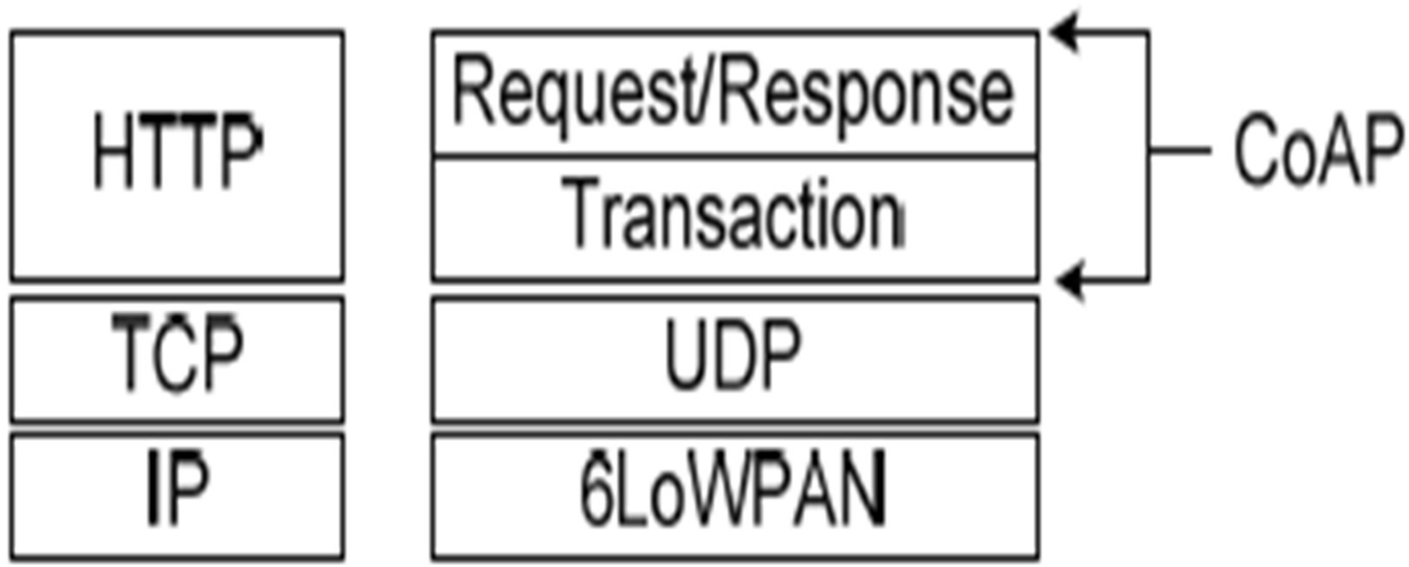


Figure 1. HTTP and CoAP protocol stacks

Design and Development of an End to End Architecture

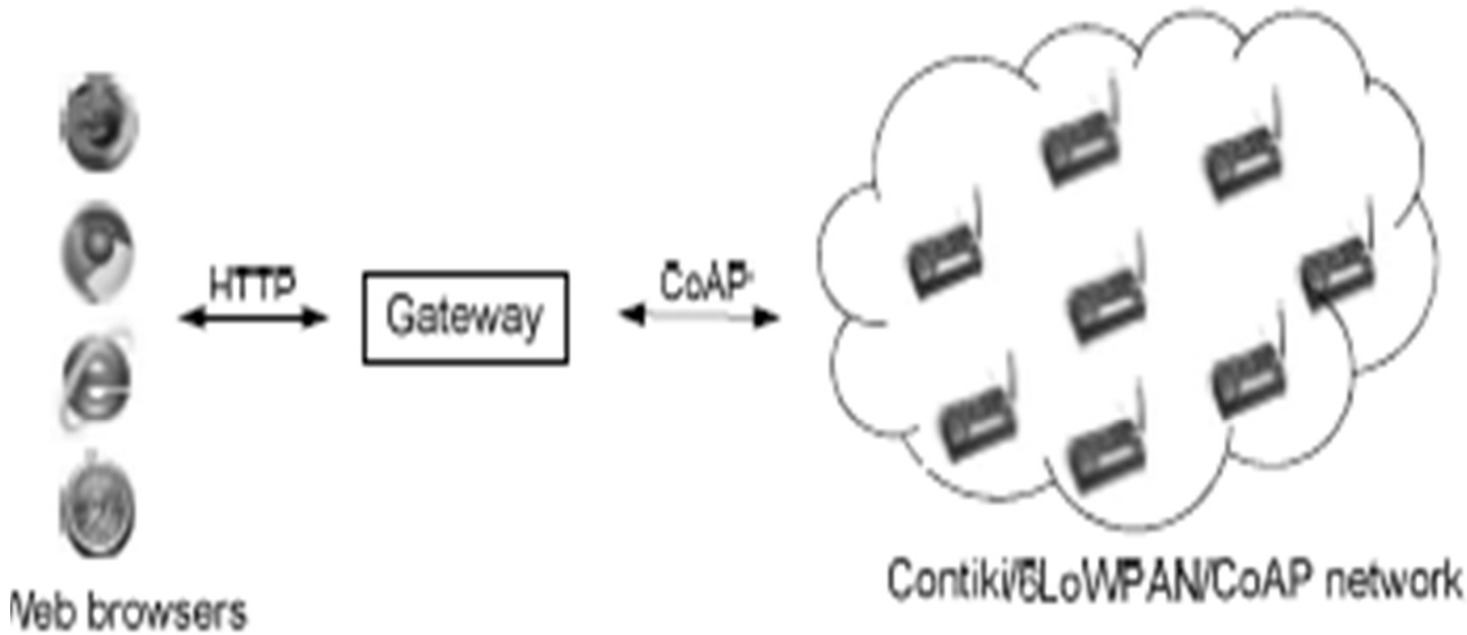
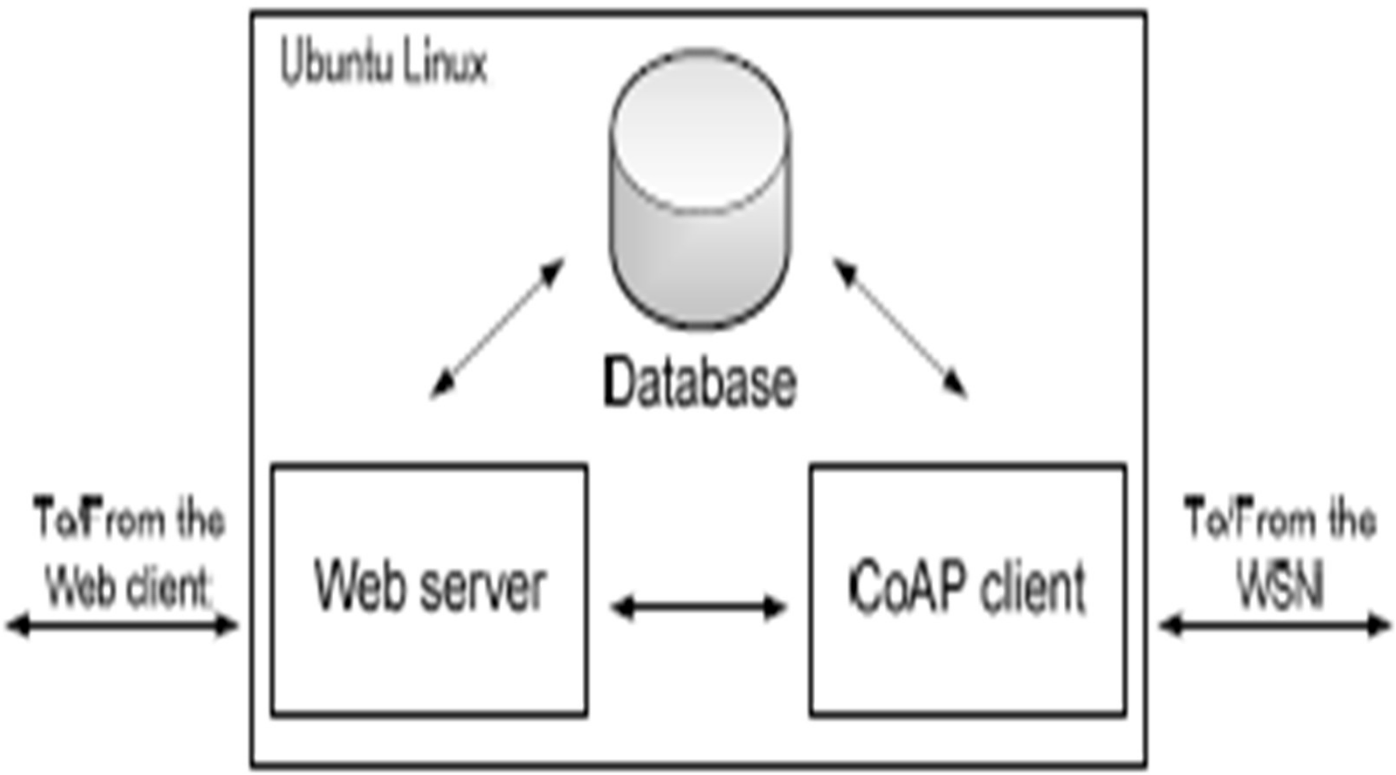
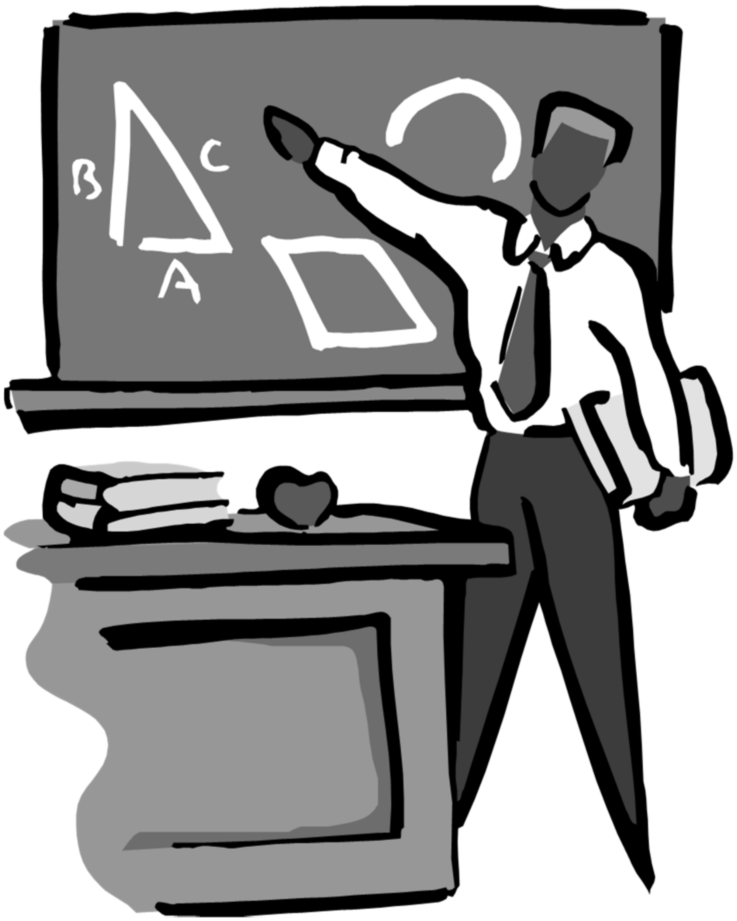


Figure 2. Integration between WSNs and the Web

Design and Development of an End to End Architecture



Cloud Based WSN Applications Provisioning



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

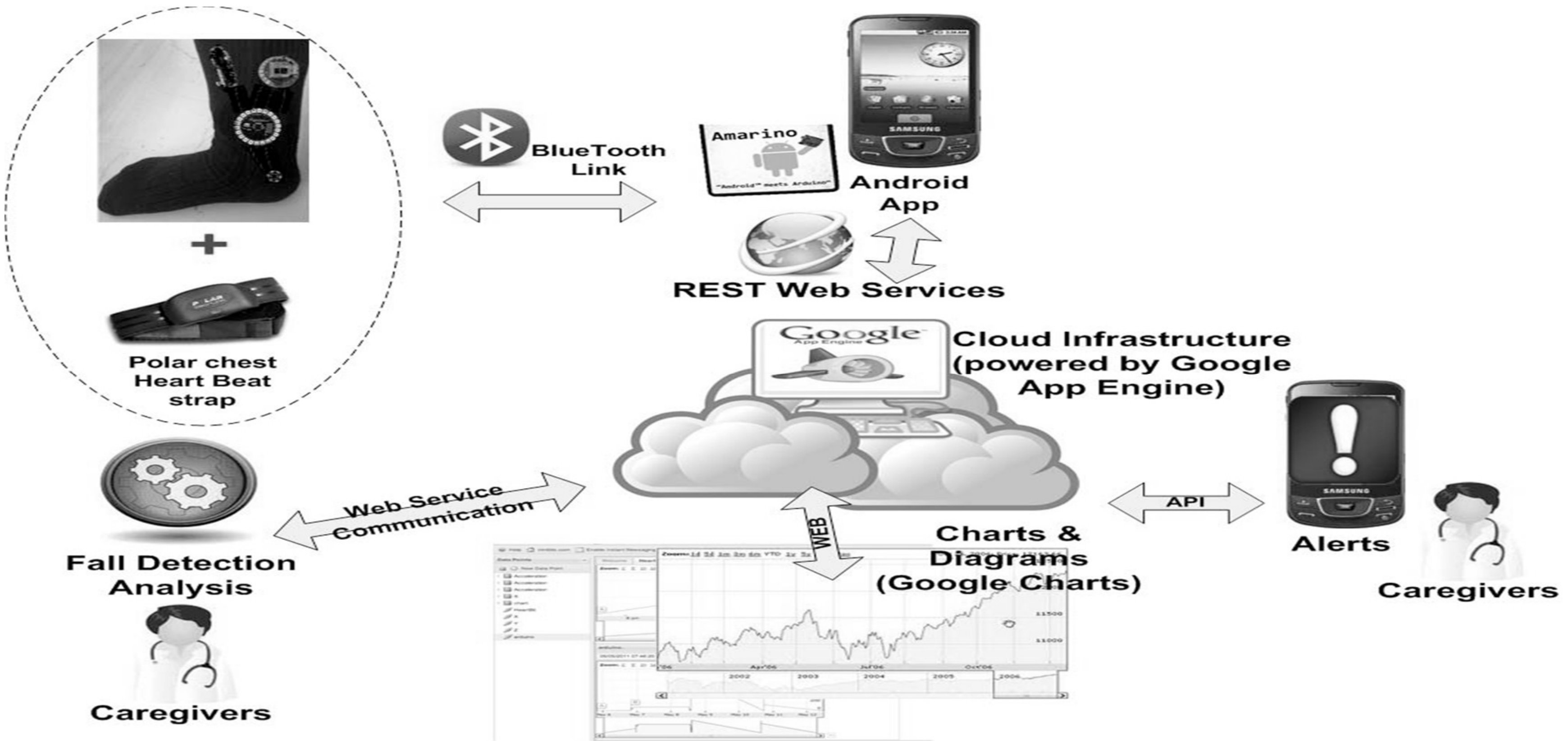
Use of Cloud Processing and Storage Power

Use of Cloud Processing and Storage Power

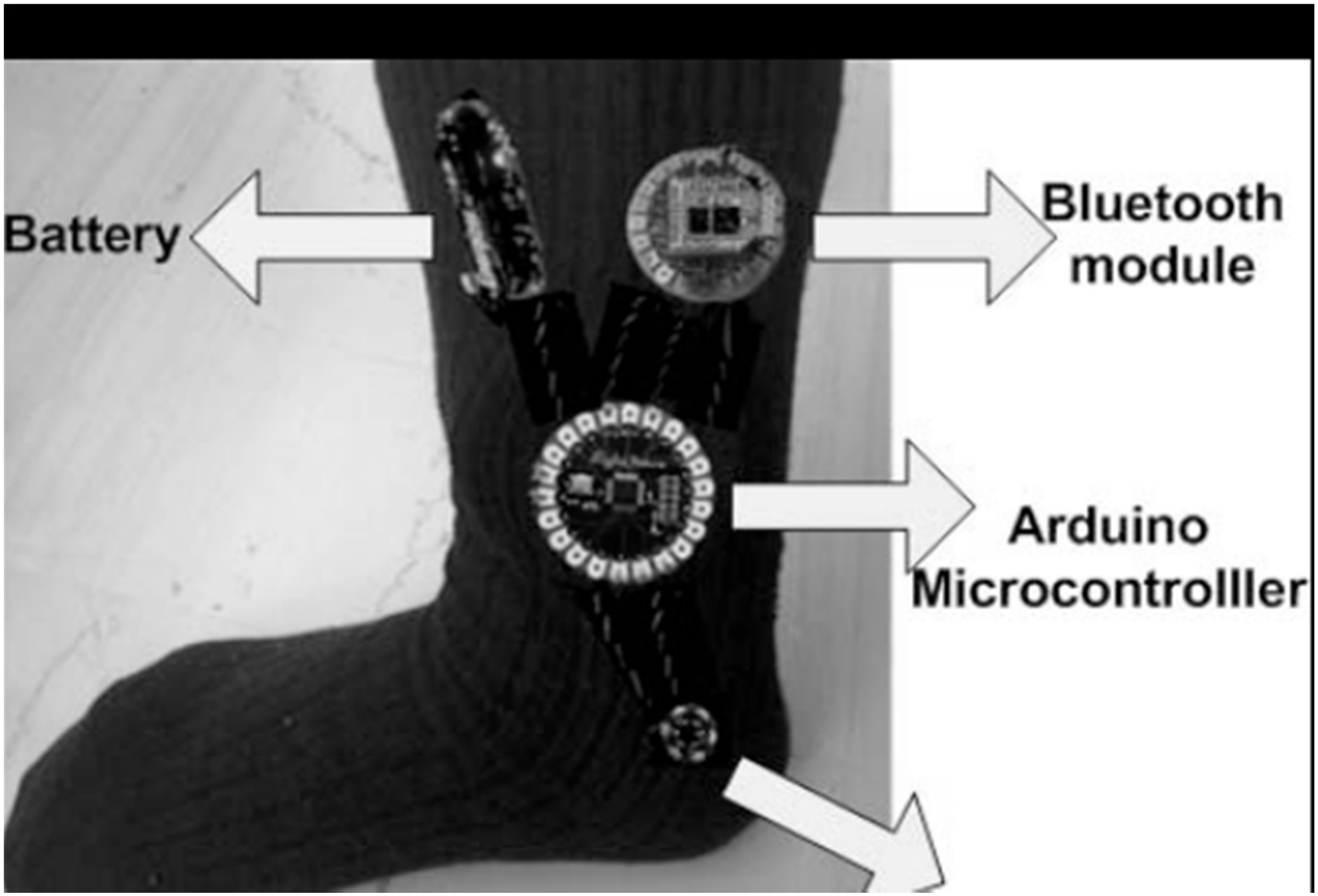
An illustration:

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

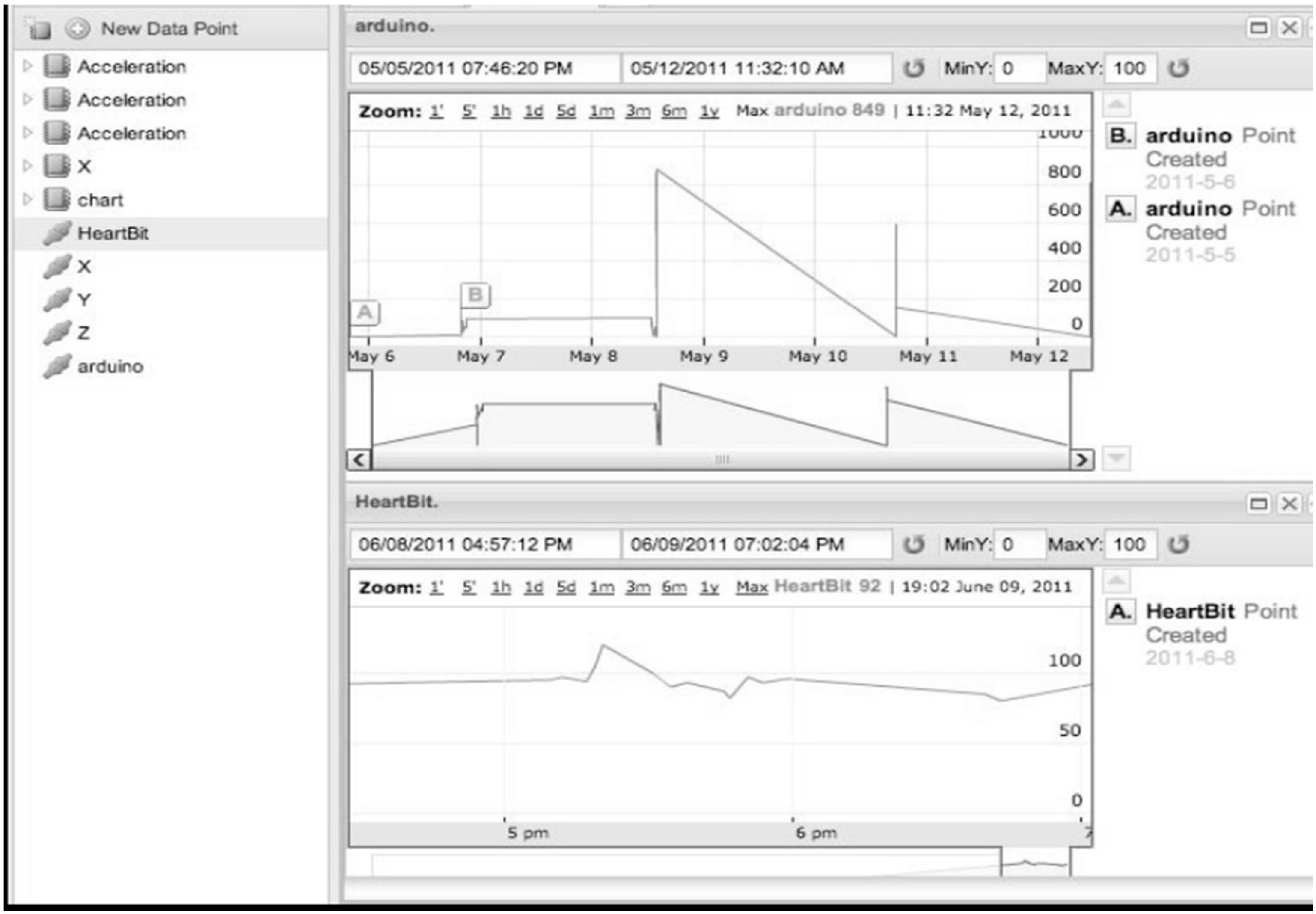
Managing Wearable Data Through Cloud



Managing Wearable Data Through Cloud



Managing Wearable Data Through Cloud



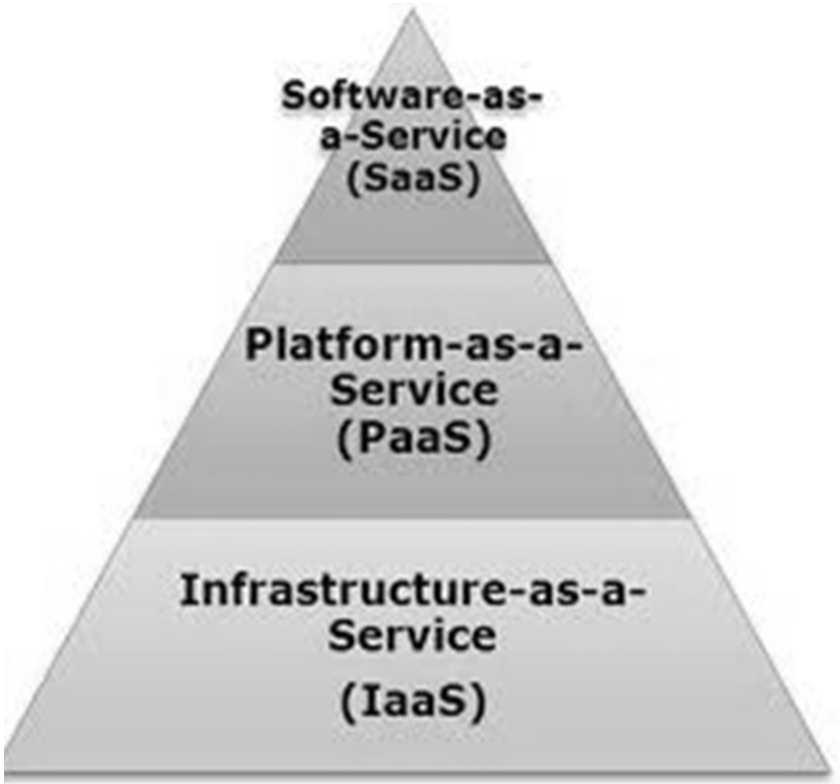
Use of Cloud Storage and Processing

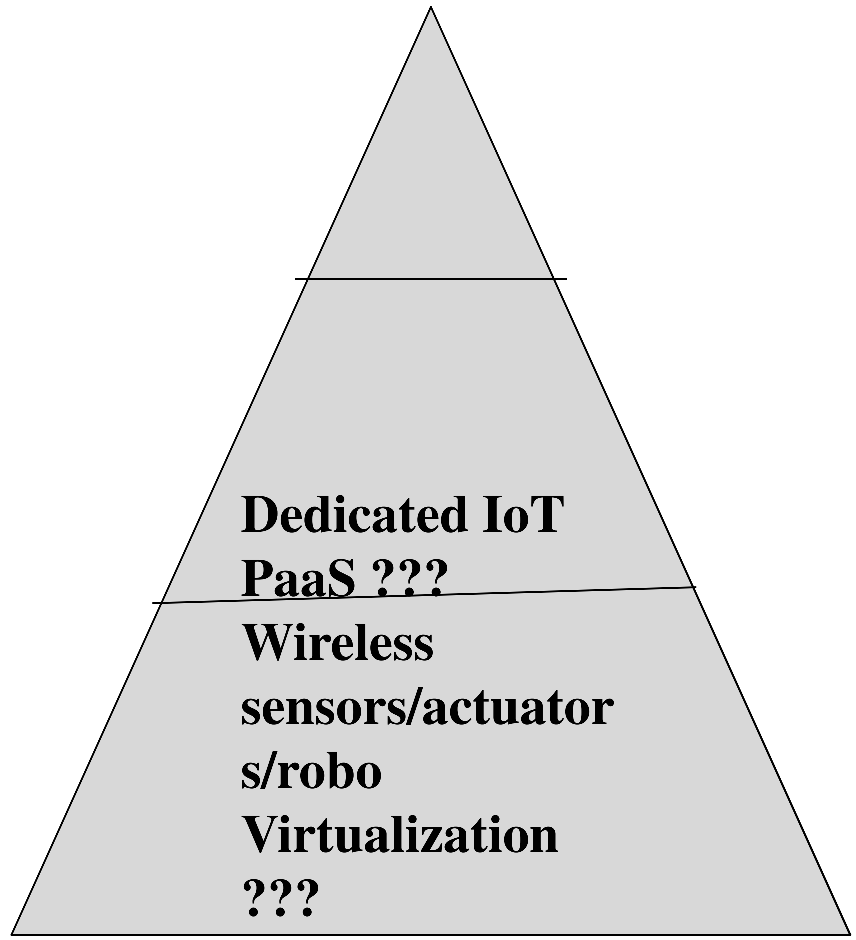
What is used from cloud computing:

- 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?

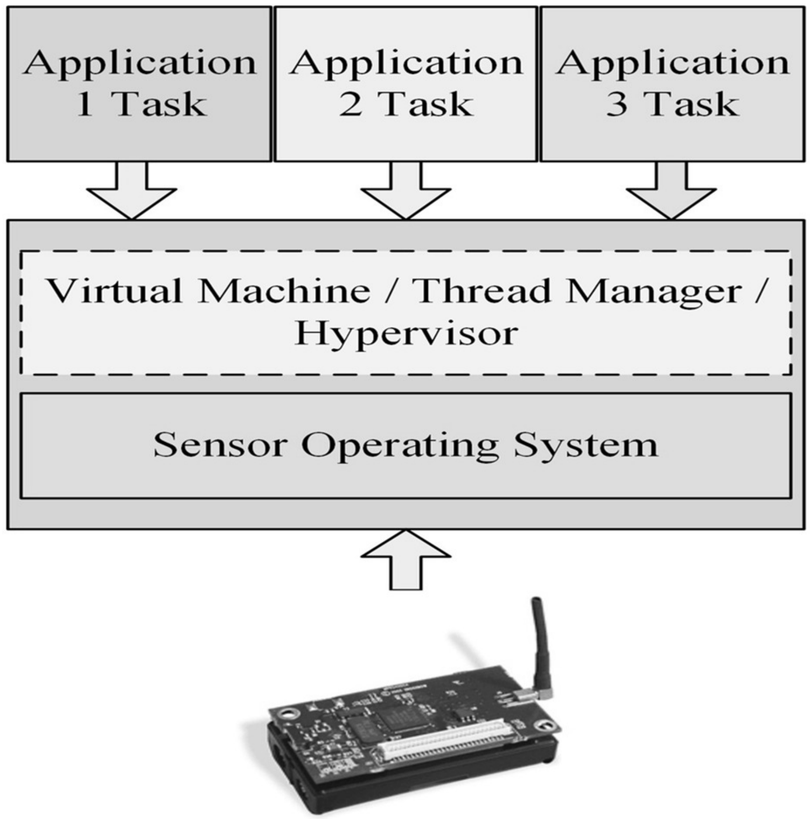
- Efficiency in resource usage through the sharing of a same WSN infrastructure by several different applications with the possibility of deploying new applications after the deployment of the WSN infrastructure

WSN Virtualization: Motivations

A scenario

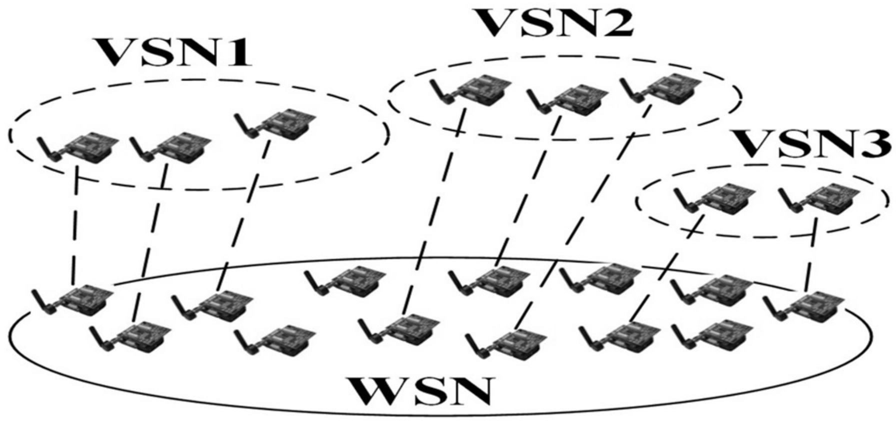
- Citizens might own a sensor in their houses to detect fire
- If the city administration decides to deploy an application which shows the fire contour (e.g. fire direction, intensity), there are 2 approaches:
  - Redeploy sensors everywhere including citizen houses
  - Deploy sensors in streets / parks 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

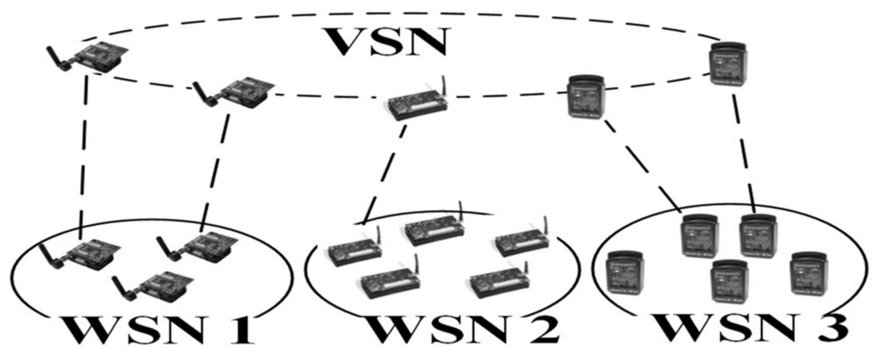


A general purpose sensor node

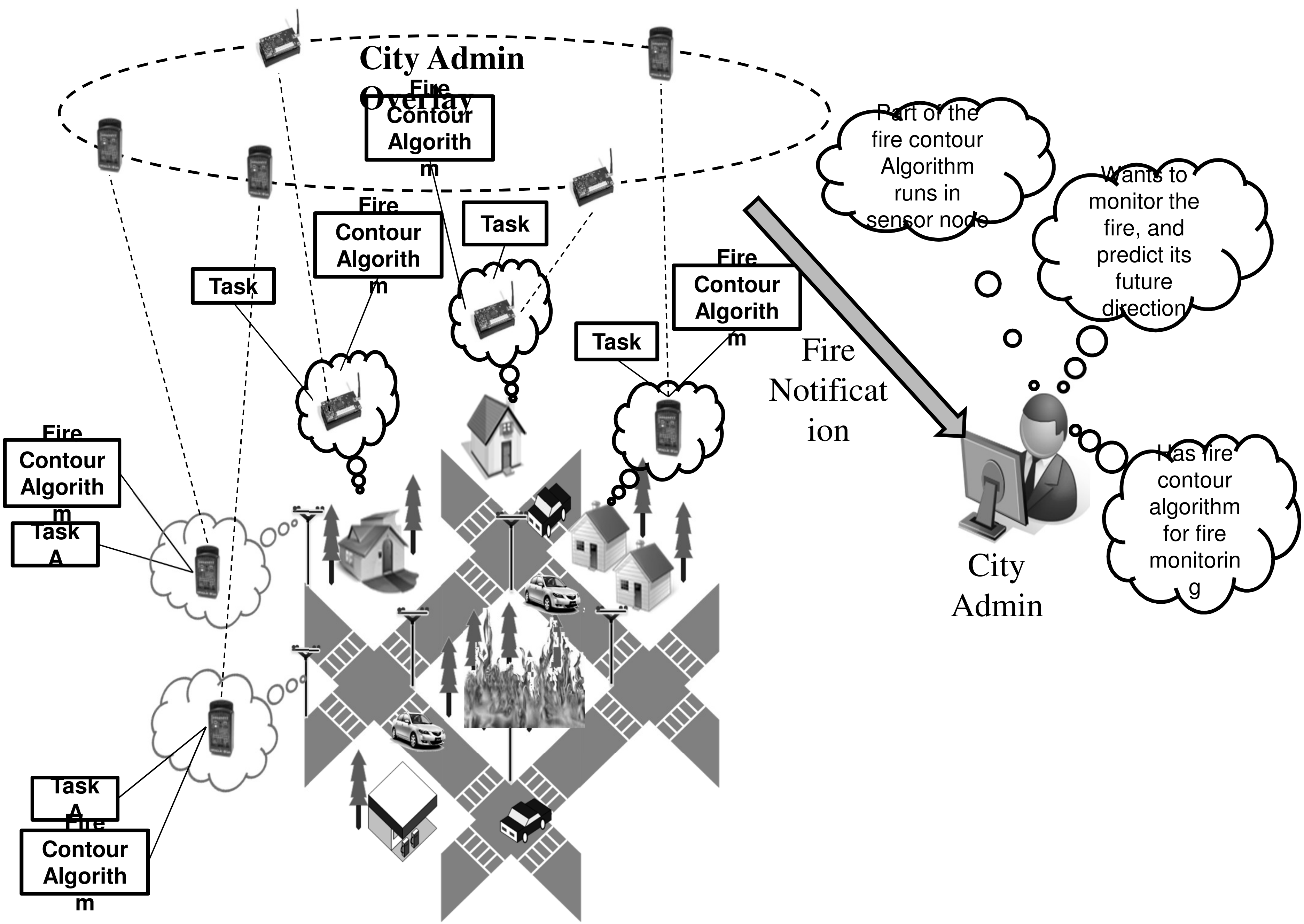
WSN Virtualization: Network Level Virtualization



a) Multiple VSNs over single WSN

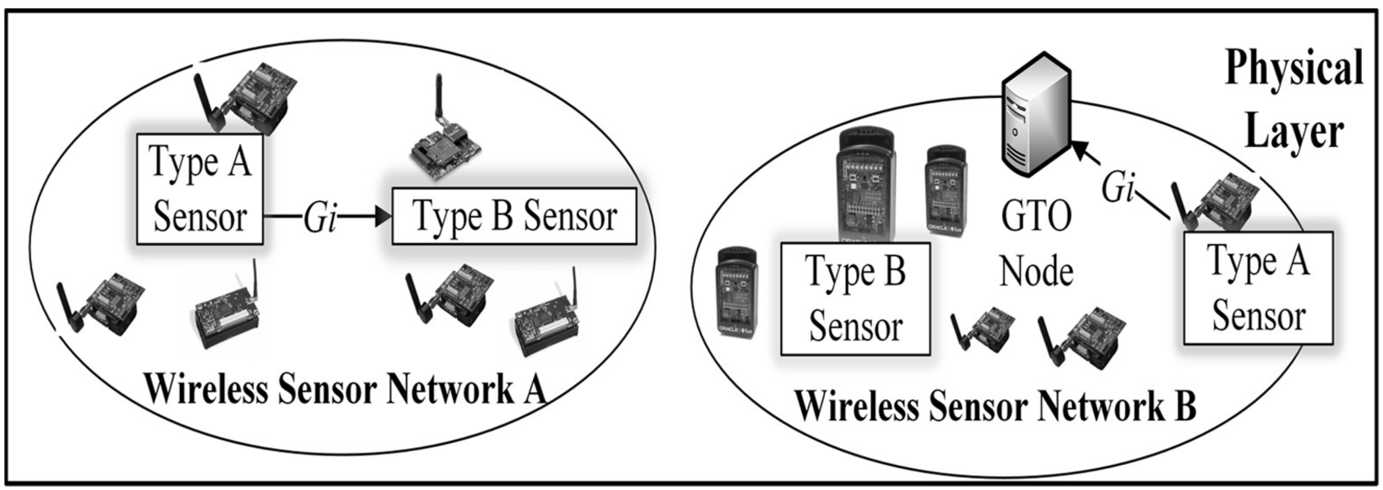


b) Single VSN over multiple WSNs


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

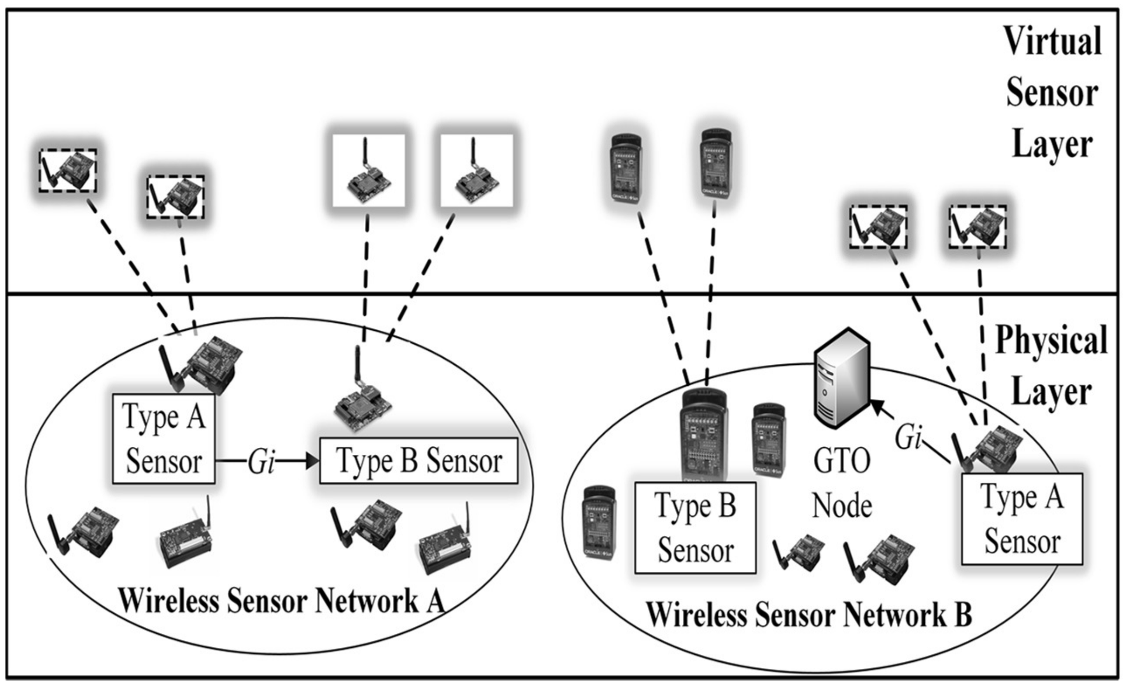
WSN Virtualization Architecture

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



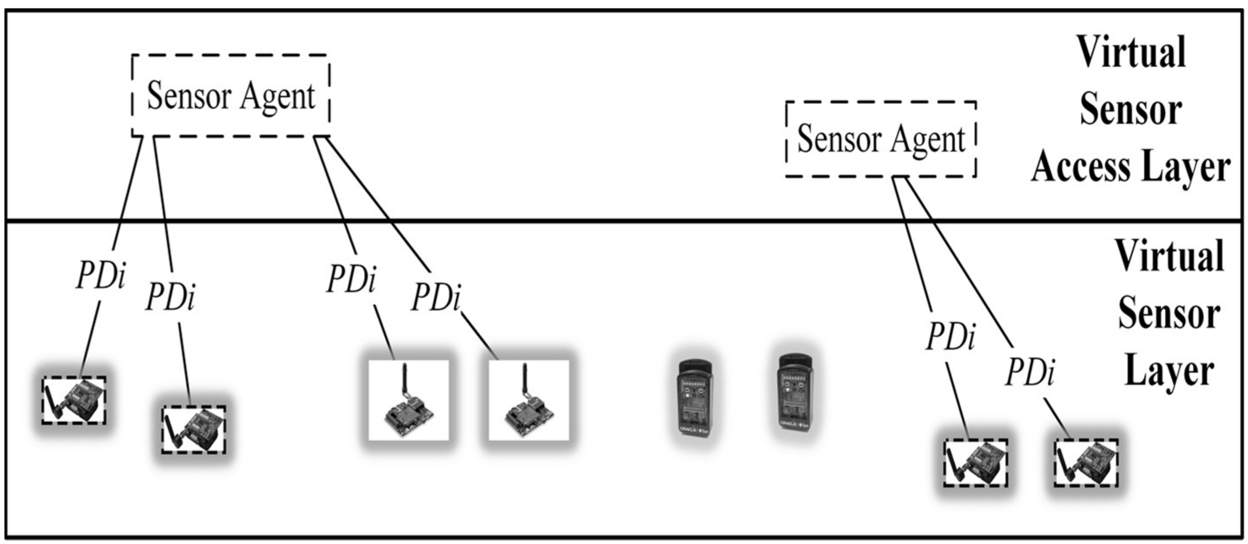
WSN Virtualization Architecture

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



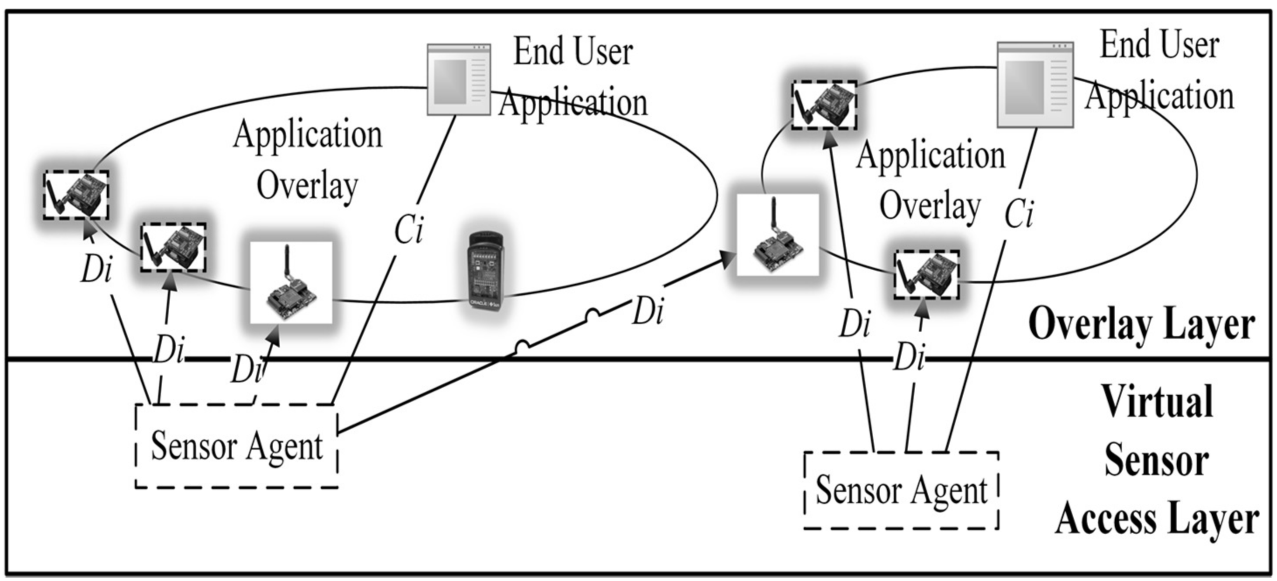
WSN Virtualization Architecture

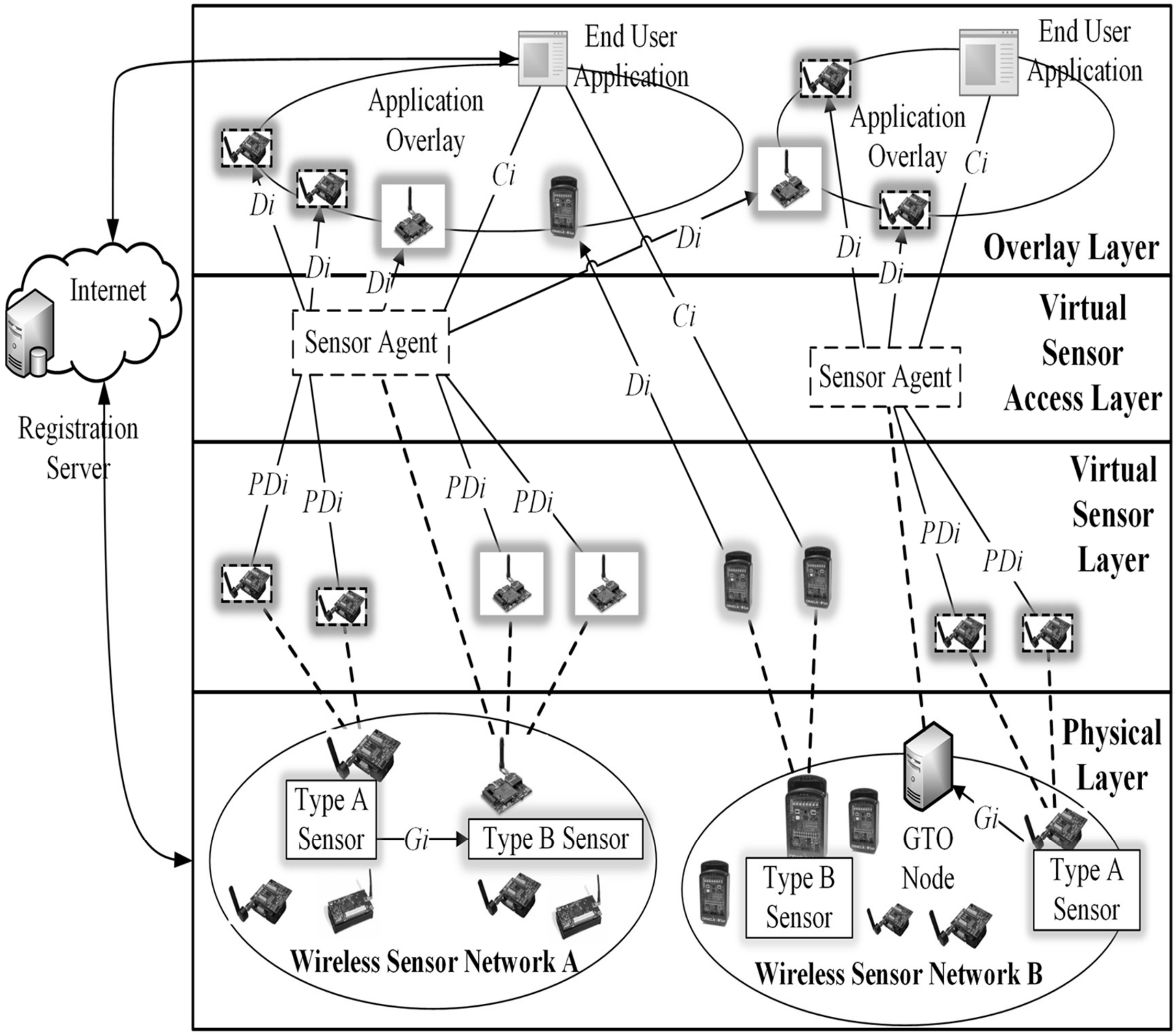
- Virtual Sensor Access Layer
  - Consists of Sensor Agents (e.g. gateways) to provide supplier/sensor brand independence
    - Abstracts virtual sensors and interacts with the applications/services



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

The End