

# WIRELESS SENSOR NETWORKS AND THEIR APPLICATION **TO** **SURVEILLANCE**: TOWARDS AMBIENT AND COOPERATIVE INTELLIGENCE

UNIVERSITÉ GASTON BERGER  
FEBRUARY 15TH, 2012  
SAINT-LOUIS, SENEGAL



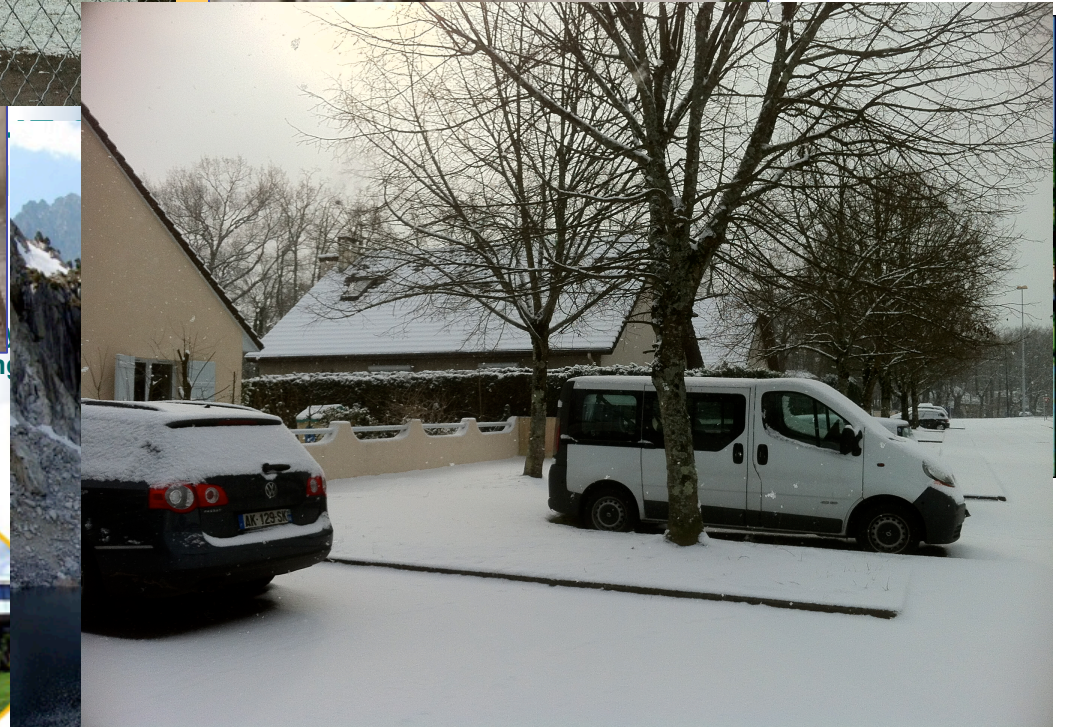
PROF. CONGDUC PHAM  
[HTTP://WWW.UNIV-PAU.FR/~CPHAM](http://www.univ-pau.fr/~cpham)  
UNIVERSITÉ DE PAU, FRANCE



# OF PAU

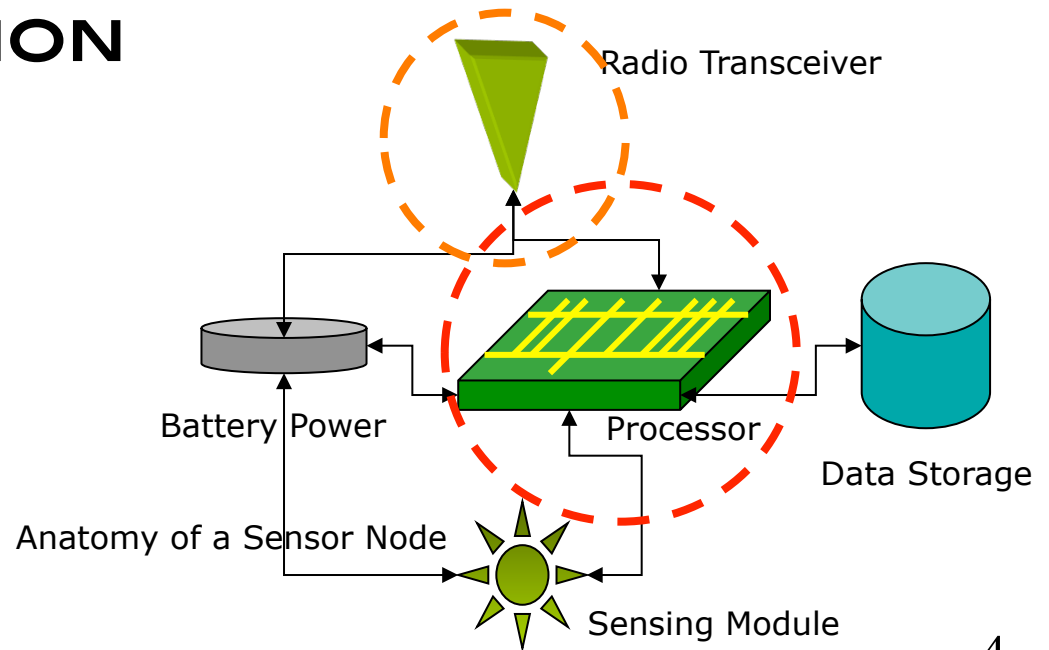
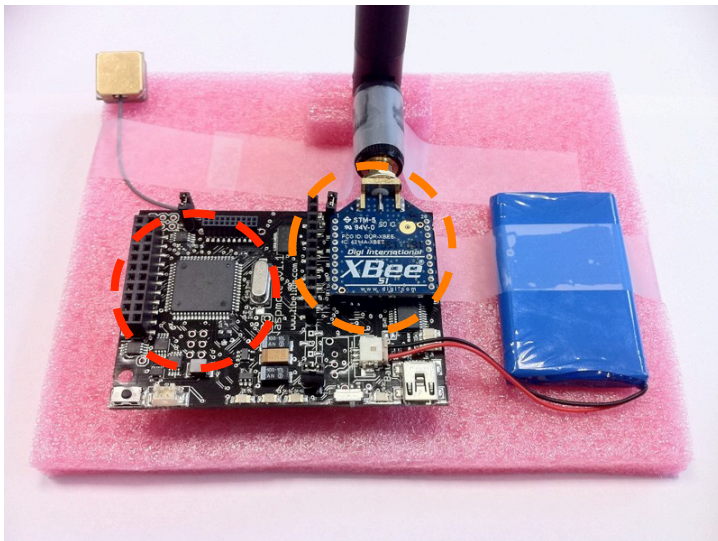


aporama des  
pus de l'UPPA



# WIRELESS AUTONOMOUS SENSORS

- ❑ IN GENERAL: LOW COST, LOW POWER (THE BATTERY MAY NOT BE REPLACEABLE), SMALL SIZE, PRONE TO FAILURE, POSSIBLY DISPOSABLE
- ❑ ROLE: SENSING, DATA PROCESSING, COMMUNICATION



# DIGITAL WIRELESS WORLD

Wi-Fi

Bluetooth

WiMax

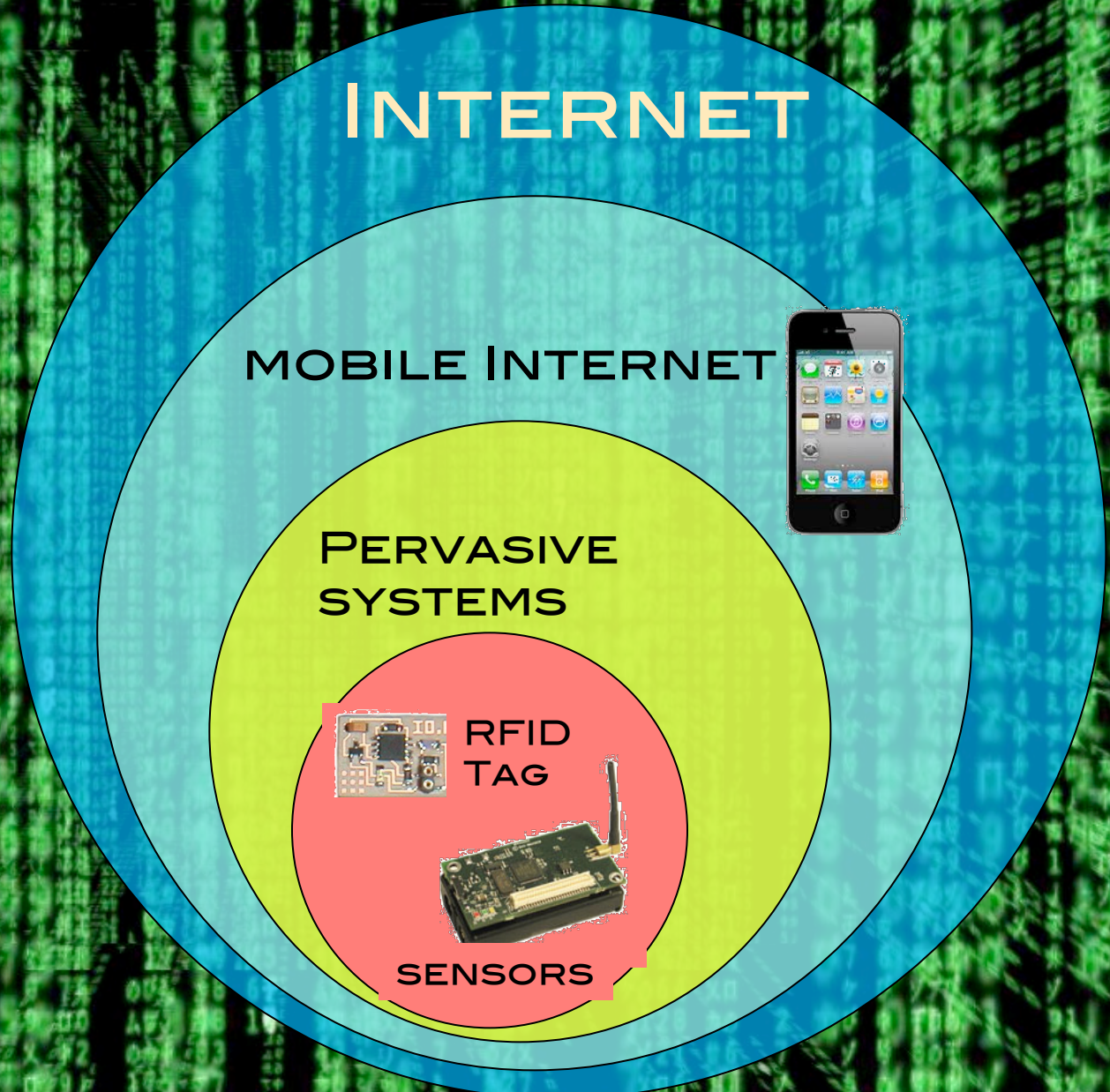
ZigBee<sup>®</sup>  
Member

4G  
100mbps-1GBps

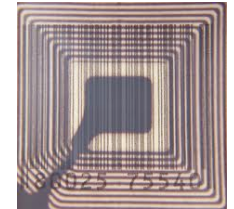
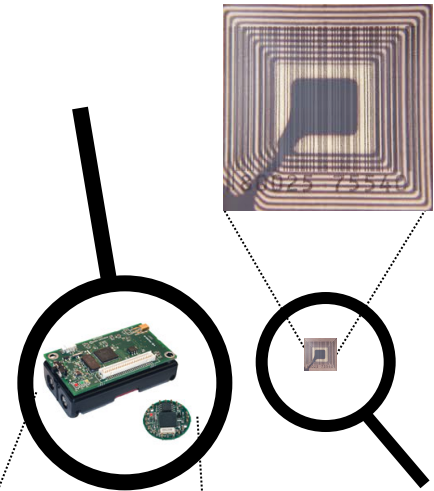
3G

Lte<sup>™</sup>

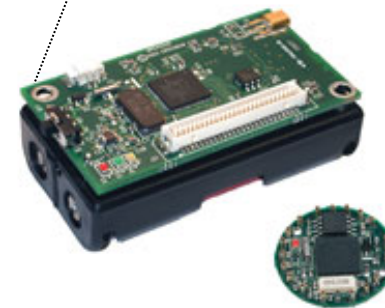
WiMedia  
ALLIANCE



# SMALL, SMART DEVICES!



Autonomous sensors and RFID tag can be embedded in various structures or objects of our daily life to enhance localization, tracking and information collection.



FROM SENSING...

SENSING



[ ...TO DIGITAL SOCIETY... ]

**PERVASIVE SYSTEMS**



**SENSING**



**[ ...TO UBIQUITOUS WORLD... ]**

**PERVASIVE SYSTEMS**



**SENSING**

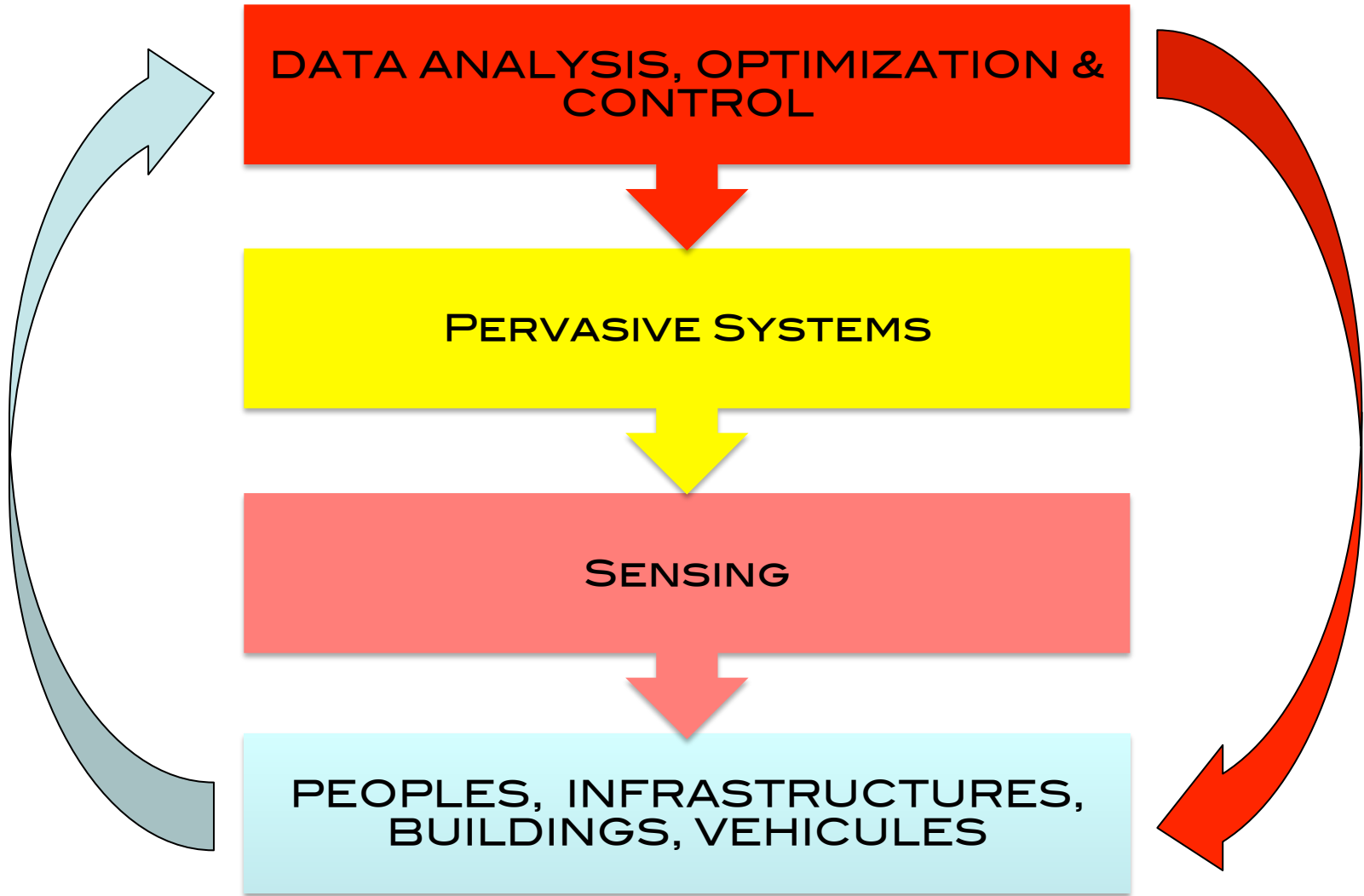


**PEOPLES,  
INFRASTRUCTURES,  
BUILDINGS, VEHICULES,...**

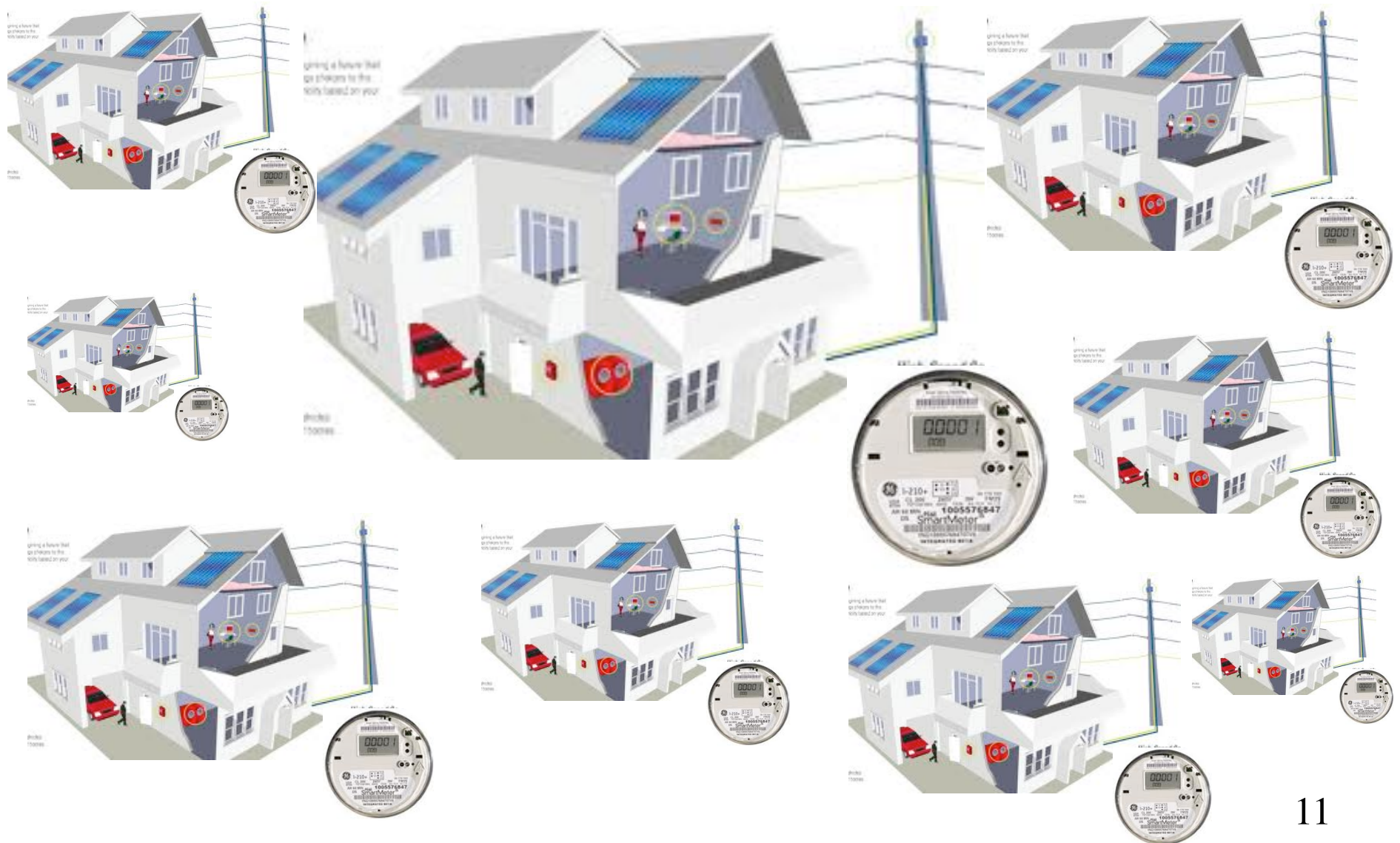




**[ ...TO CONTROLLED SYSTEMS. ]**



# EX: SMART ELECTRICITY NETWORKS

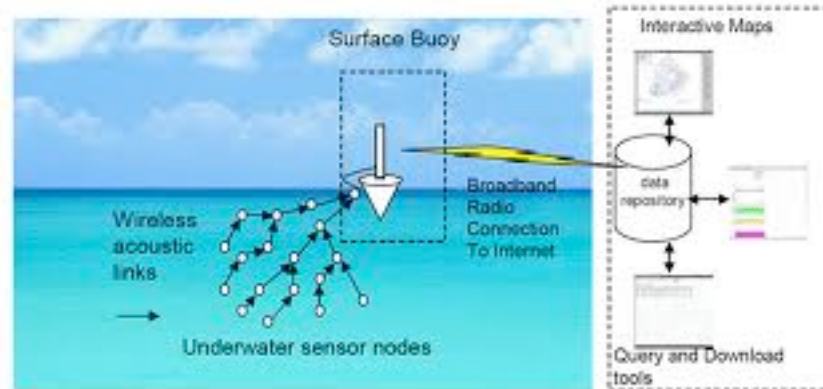


# EX: SMART ELECTRICITY NETWORKS



Yogesh Simmhan, Baohua Cao, Michail Giakkoupis, and Viktor K. Prasanna. **Adaptive rate stream processing for smart grid applications on clouds**. In Proceedings of the 2nd ACM international workshop on Scientific cloud computing (ScienceCloud '11).

# MONITORING/SURVEILLANCE



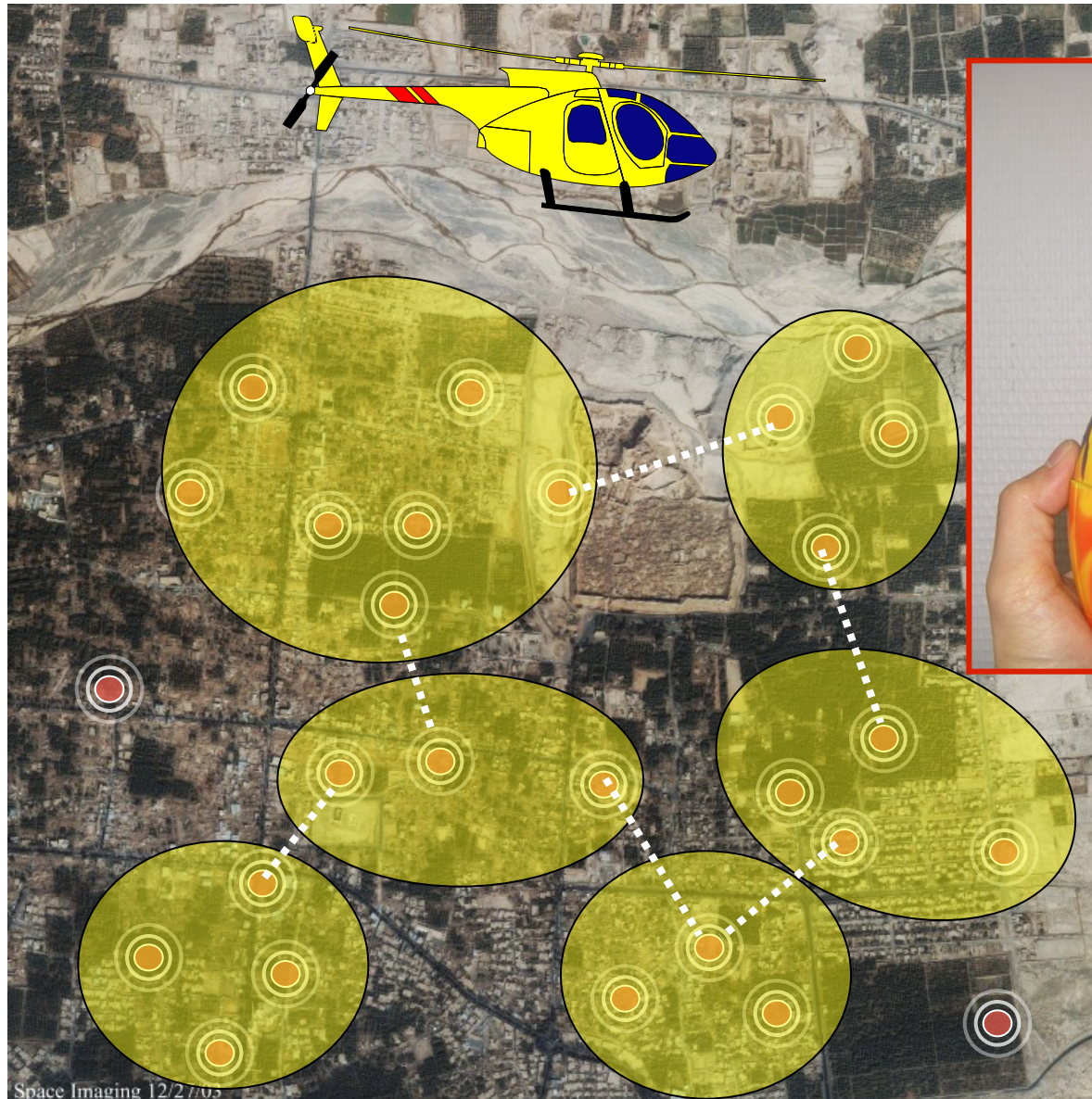
# SEARCH&RESCUE, SECURITY



Imote2



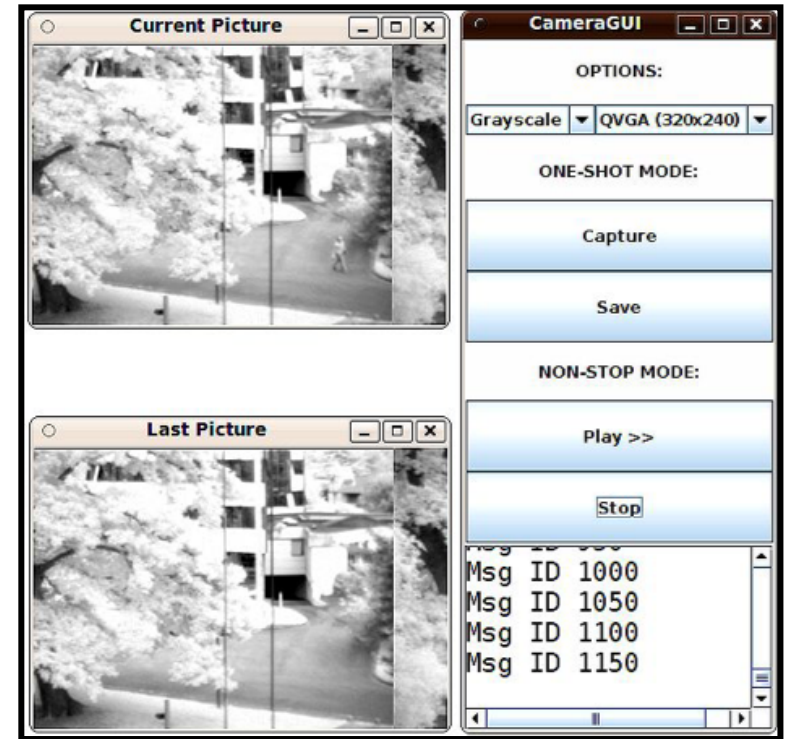
Multimedia board



# CROSSBOW MOTES OF OUR TESTBED

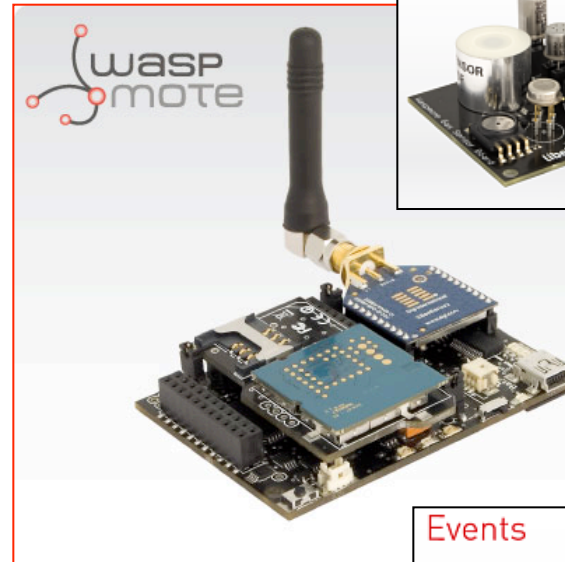


iMot



iMote2 with IMB400  
multimedia board

- ❑ ATMEGA1281 MICROCONTROLLER
- ❑ 8K RAM & 1G SD CARD.
- ❑ 2.4GHZ IEEE 802.15.4 COMPATIBLE. RF AND GSM/GPRS



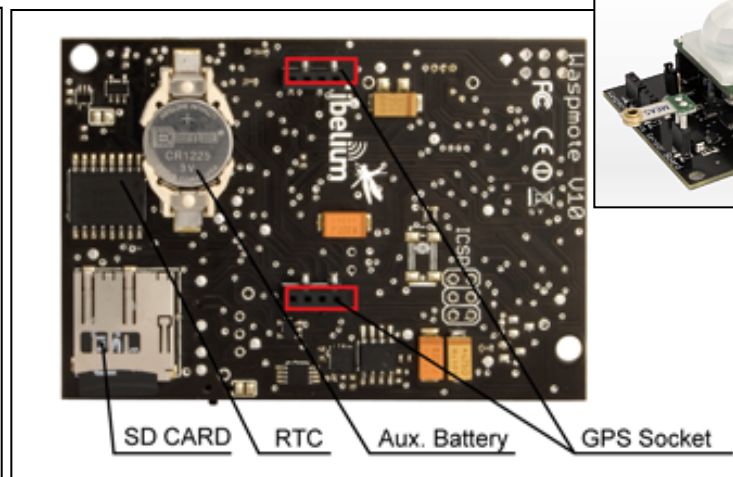
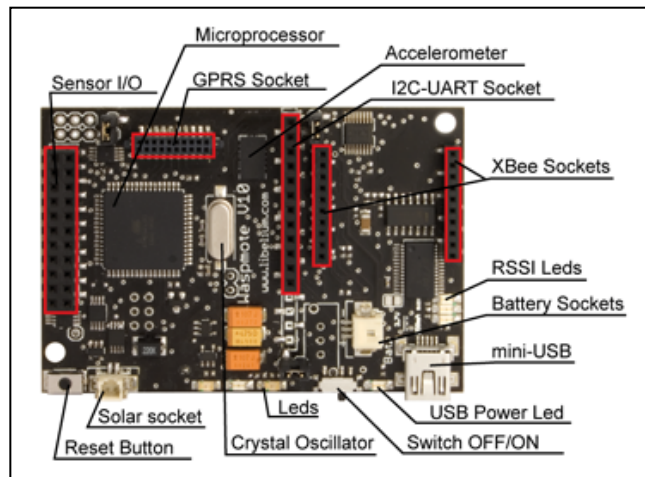
Gases

- Carbon Monoxide – CO
- Carbon Dioxide – CO2
- Oxygen – O2
- Methane – CH4
- Hydrogen – H2
- Ammonia – NH3
- Isobutane – C4H10
- Ethanol – CH3CH2OH
- Toluene – C6H5CH3
- Hydrogen Sulfide – H2S
- Nitrogen Dioxide – NO2
- Temperature
- Humidity

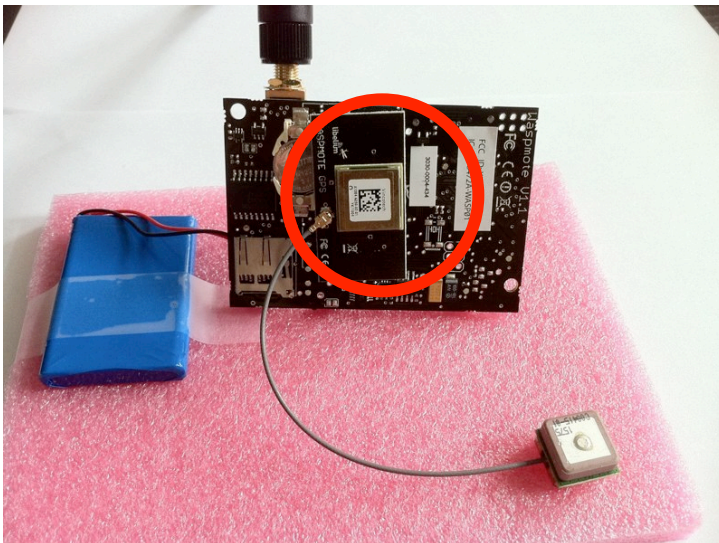
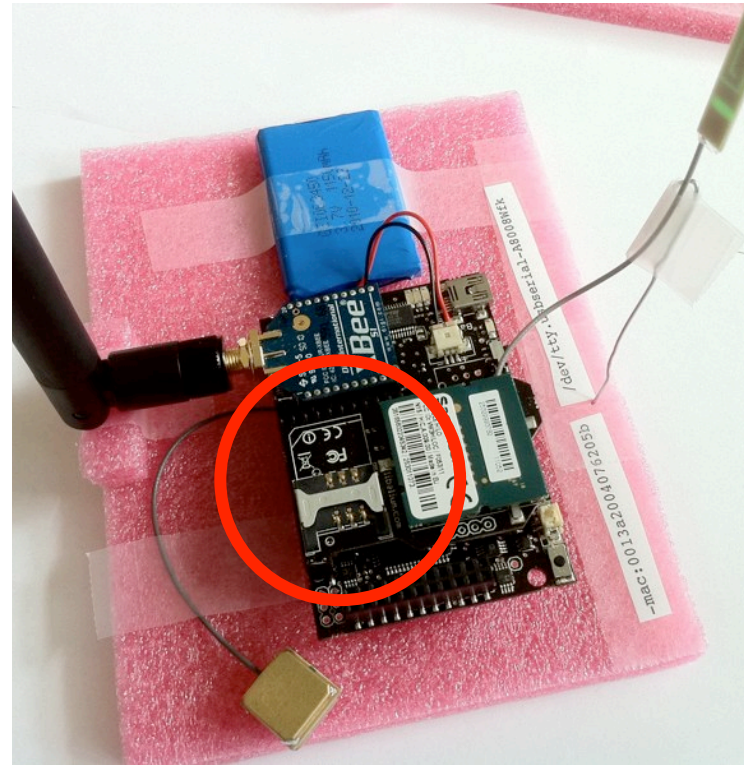
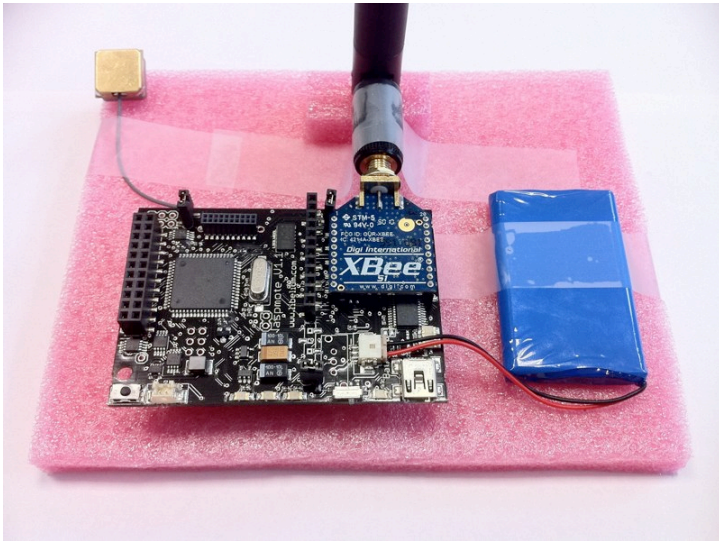


Events

- Pressure/Weight
- Bend
- Vibration
- Impact
- Hall Effect
- Tilt
- Temperature (+/-)
- Liquid Presence
- Liquid Level
- Luminosity
- Presence (PIR)
- Stretch

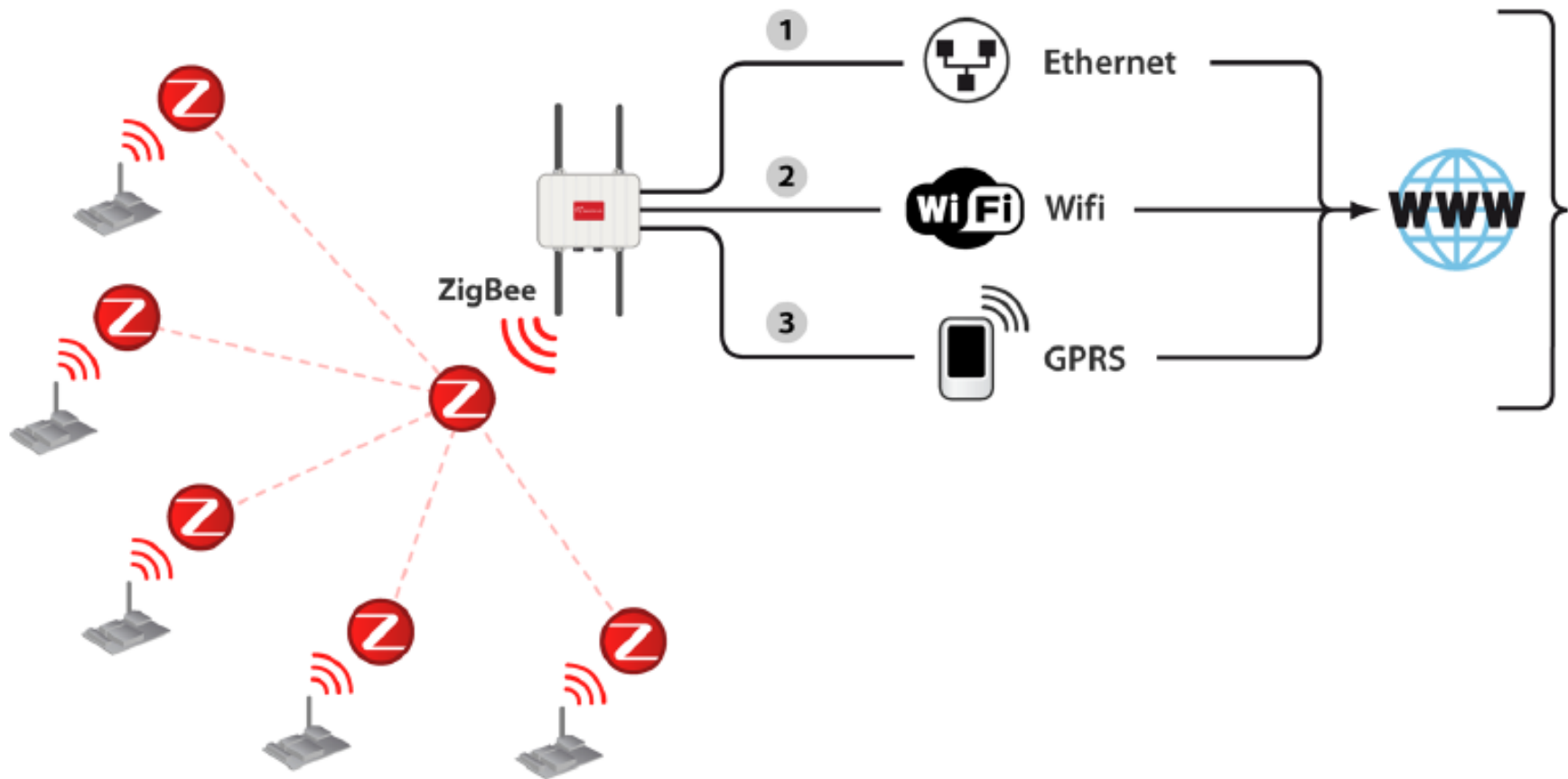


# LIBELIUM WASPMOTE (1)

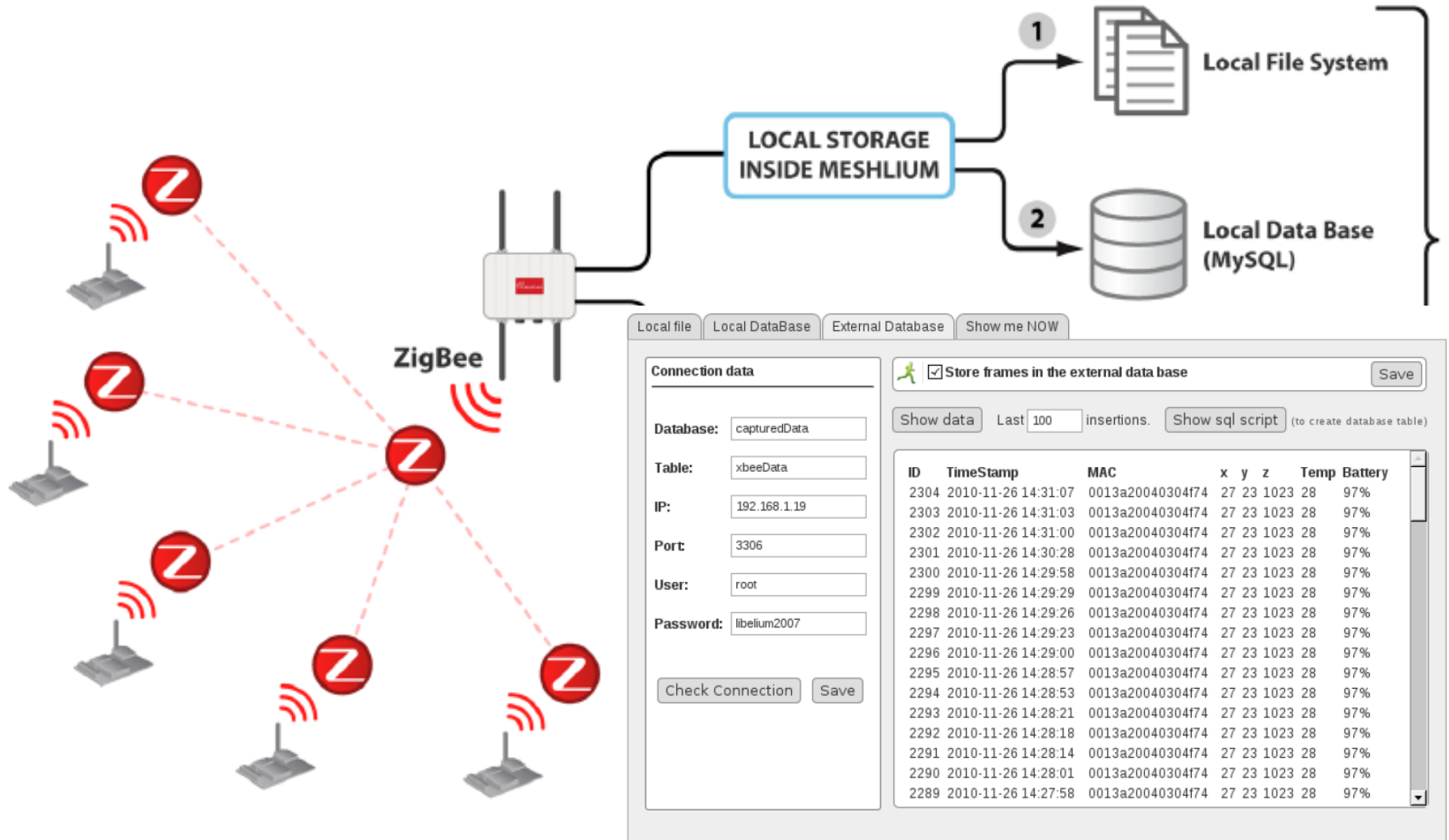




# ADVANCED CONNECTIVITY



# ADVANCED DATABASE FEATURES



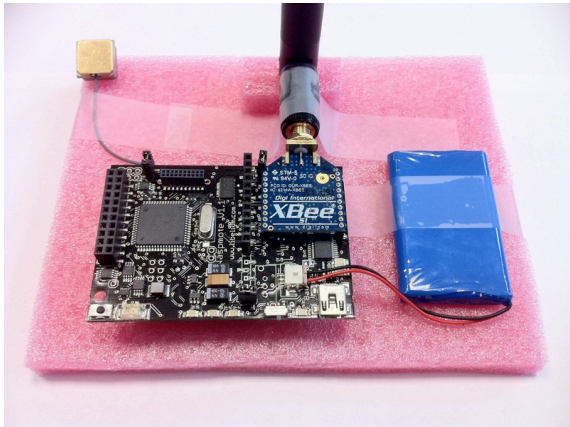
# THE FULL TESTBED



# CUSTOM BEHAVIOR

```
void setup()  
{  
  ACC.ON();  
  USB.begin(); // starts u  
}
```

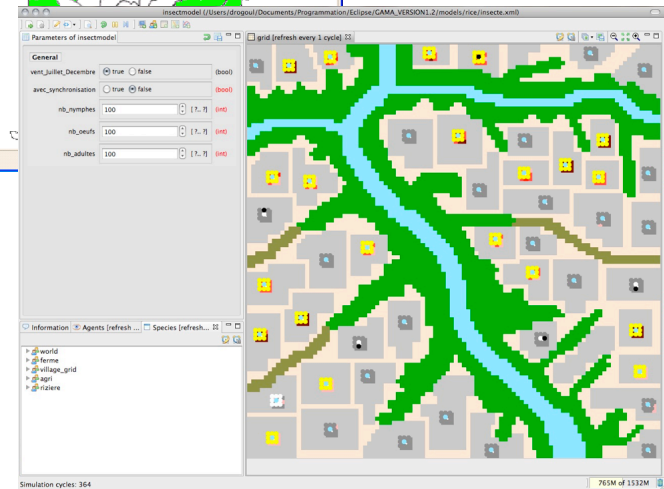
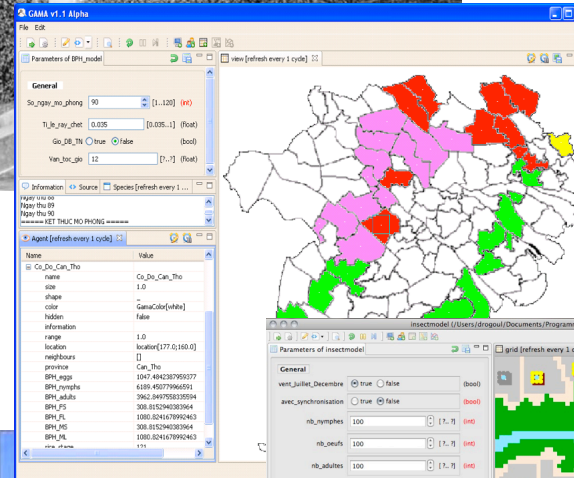
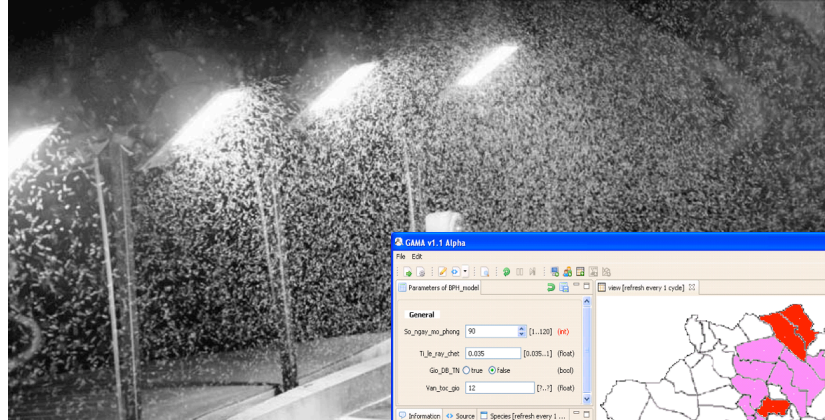
```
void loop()  
{  
  
  //-----Check Register-----  
  // should always answer 0x3A, it is used to check  
  // the proper functionality of the accelerometer  
  byte check = ACC.check();  
  
  //-----X Values-----  
  int x_acc = ACC.getX();  
  
  //-----Y Values-----  
  int y_acc = ACC.getY();  
  
  //-----Z Values-----  
  int z_acc = ACC.getZ();  
  
  //-----
```



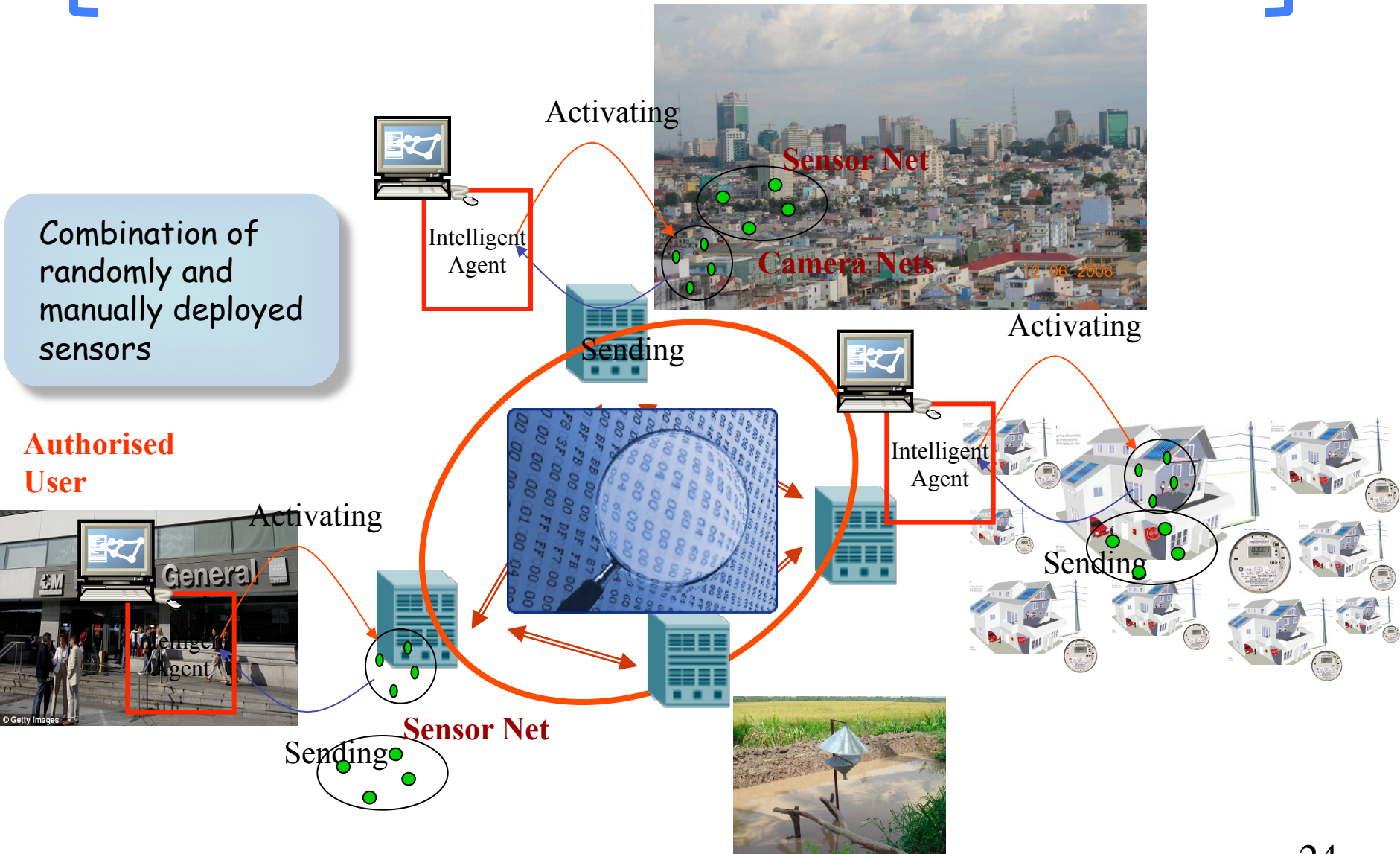
# AIR QUALITY MONITORING



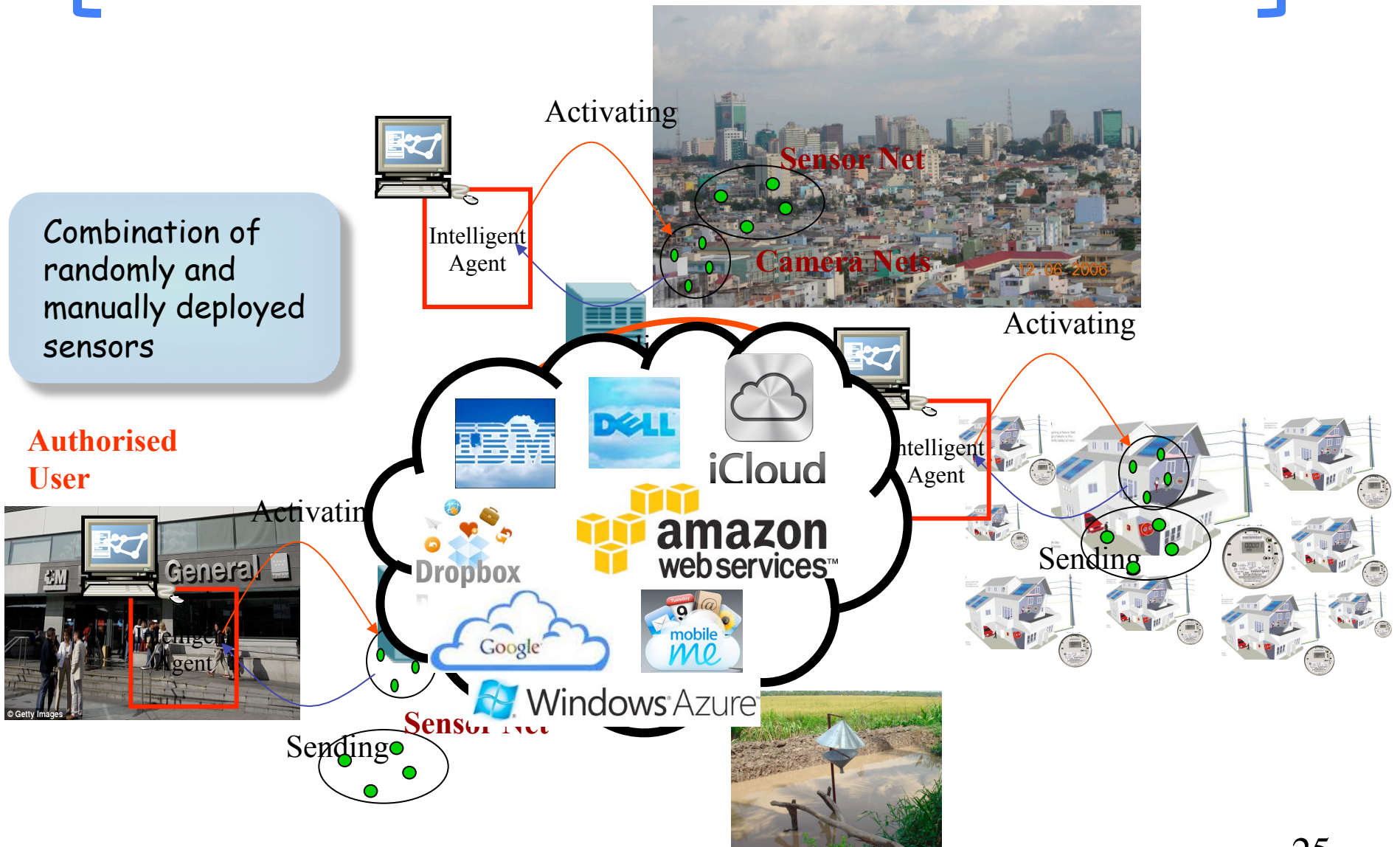
# SPECIFIC APPLICATIONS



# TOWARDS GLOBAL SENSING

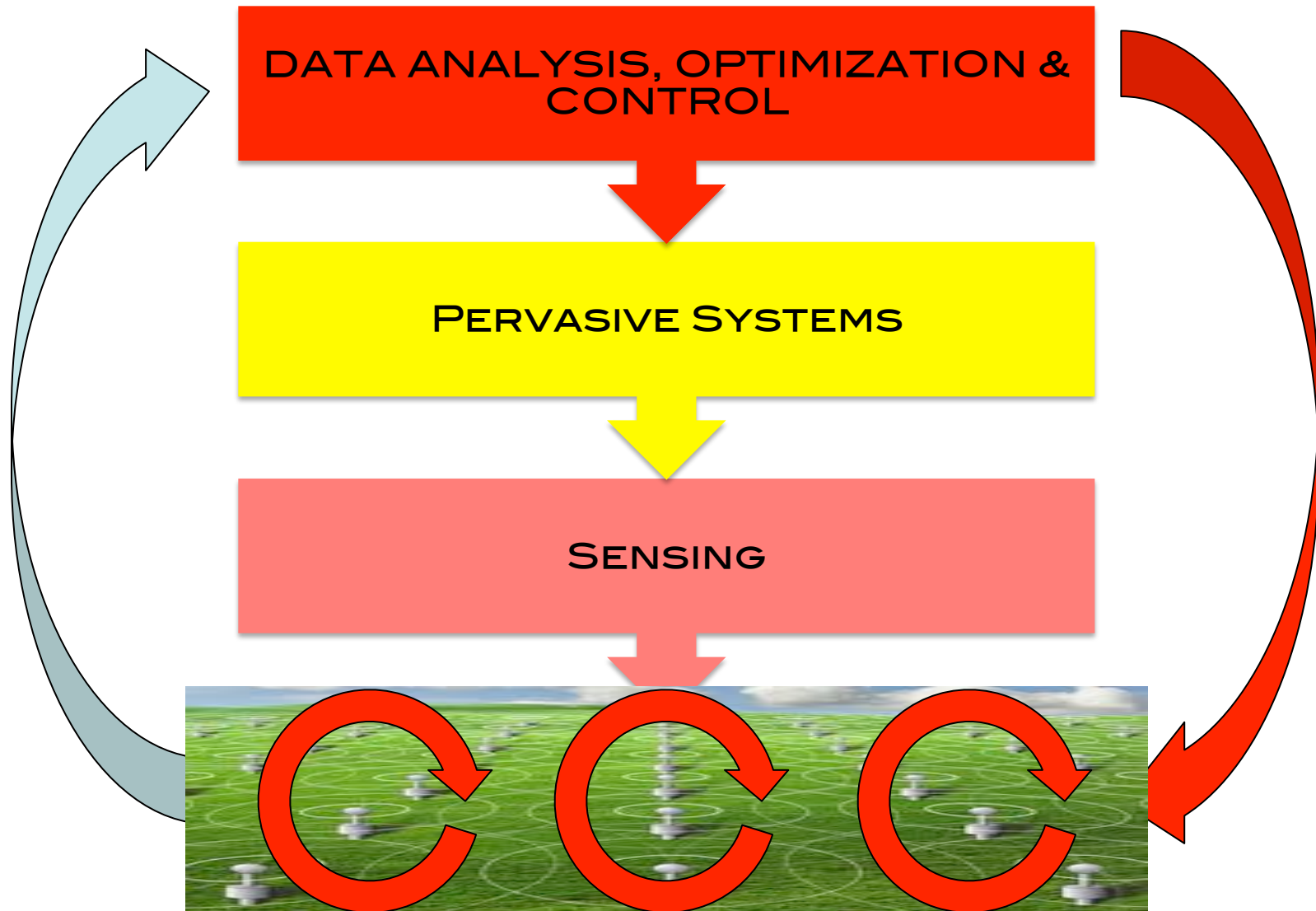


# WHERE CLOUDS COME IN!

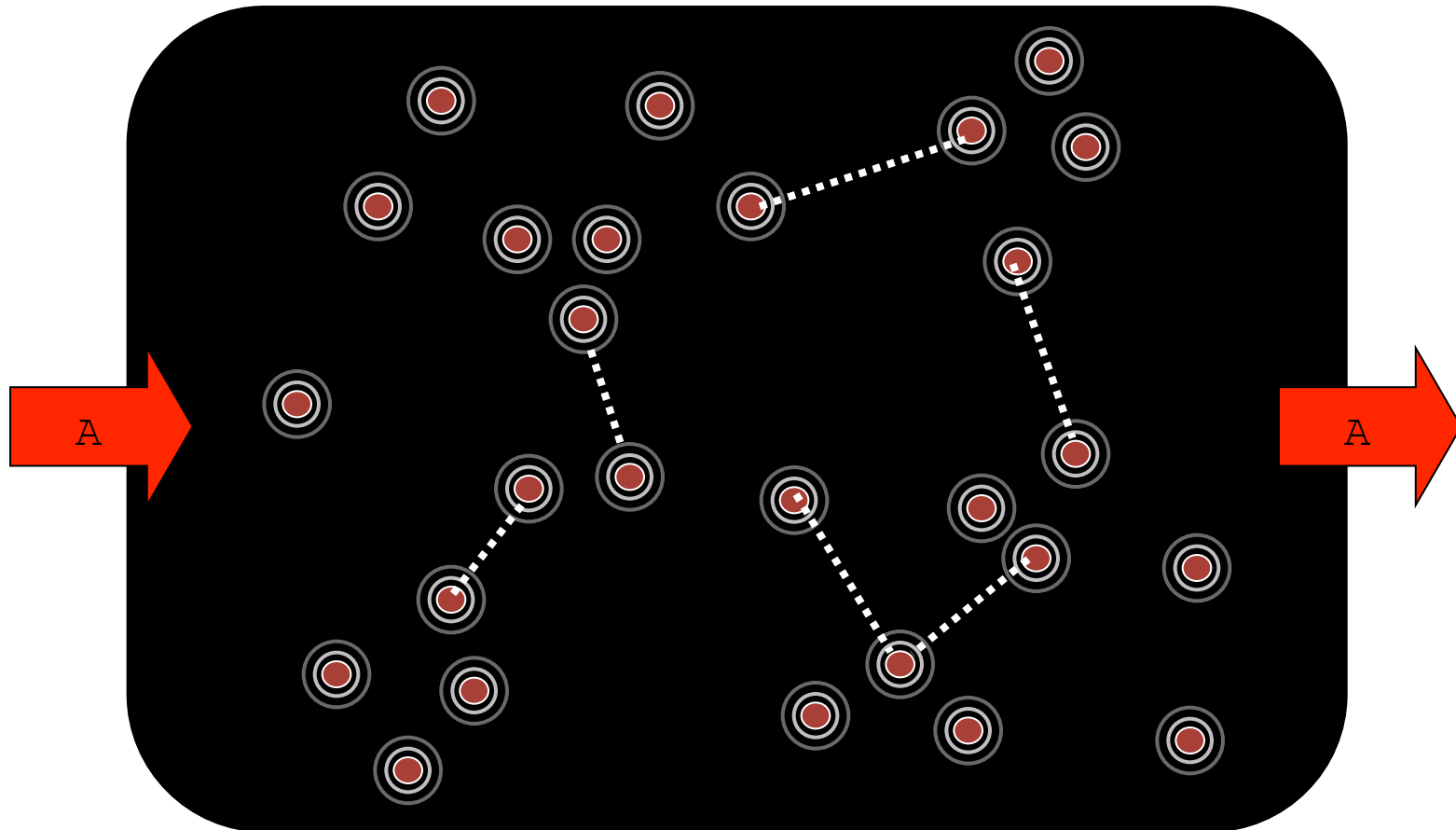




# AMBIENT INTELLIGENCE?

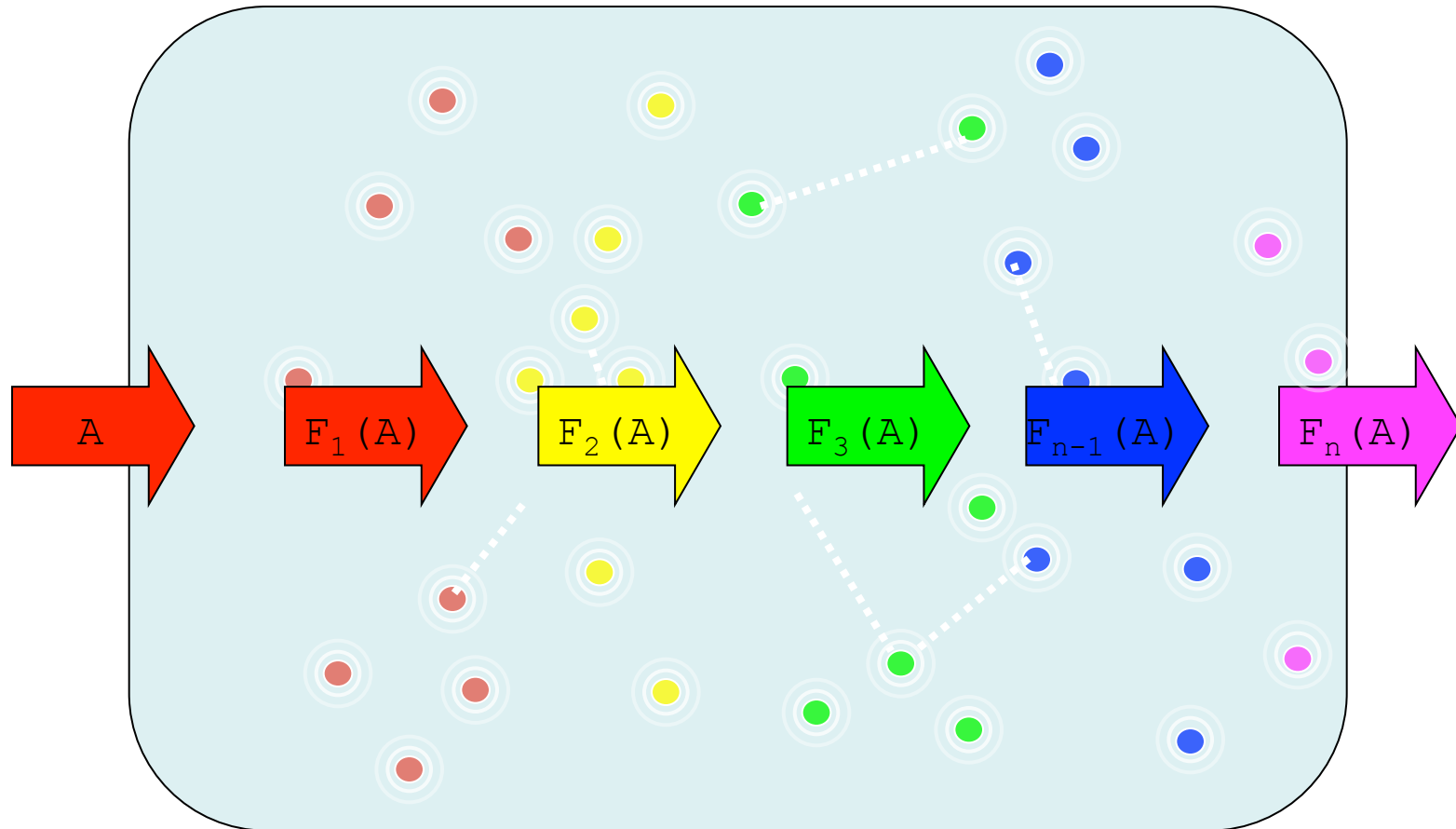


# ENABLING AMBIENT INTELLIGENCE (1)



FROM A BLACK-BOX  
VISION...

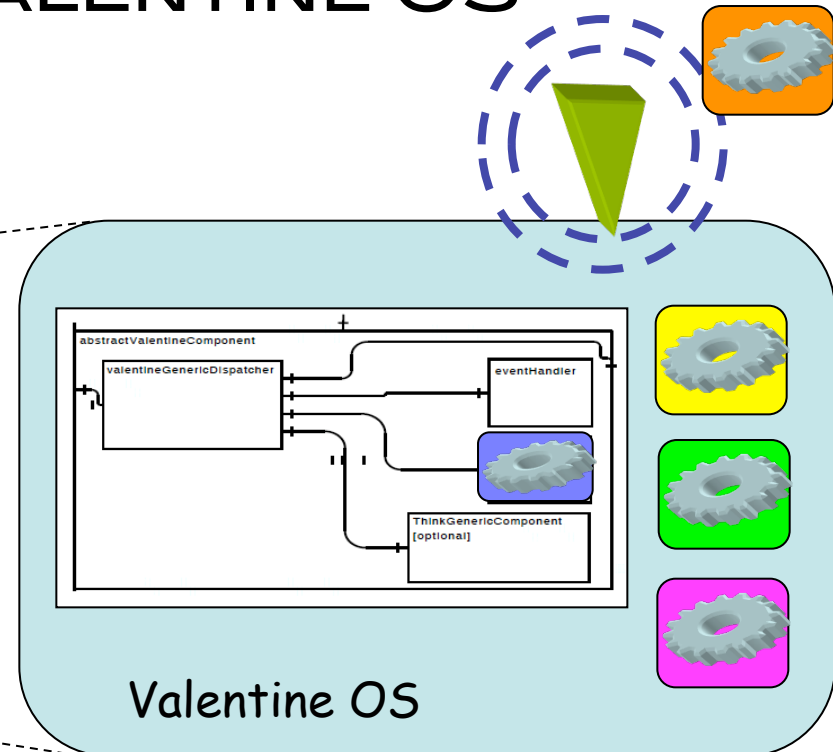
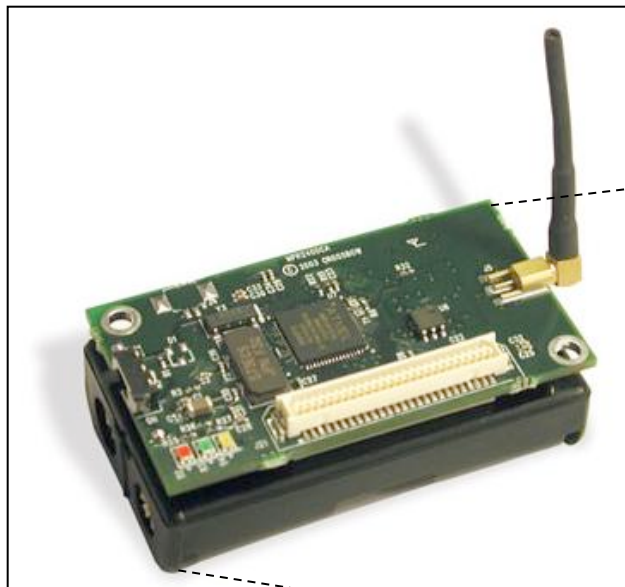
# ENABLING AMBIENT INTELLIGENCE (2)



...TO A DYNAMIC, ADAPTIVE  
BEHAVIOR

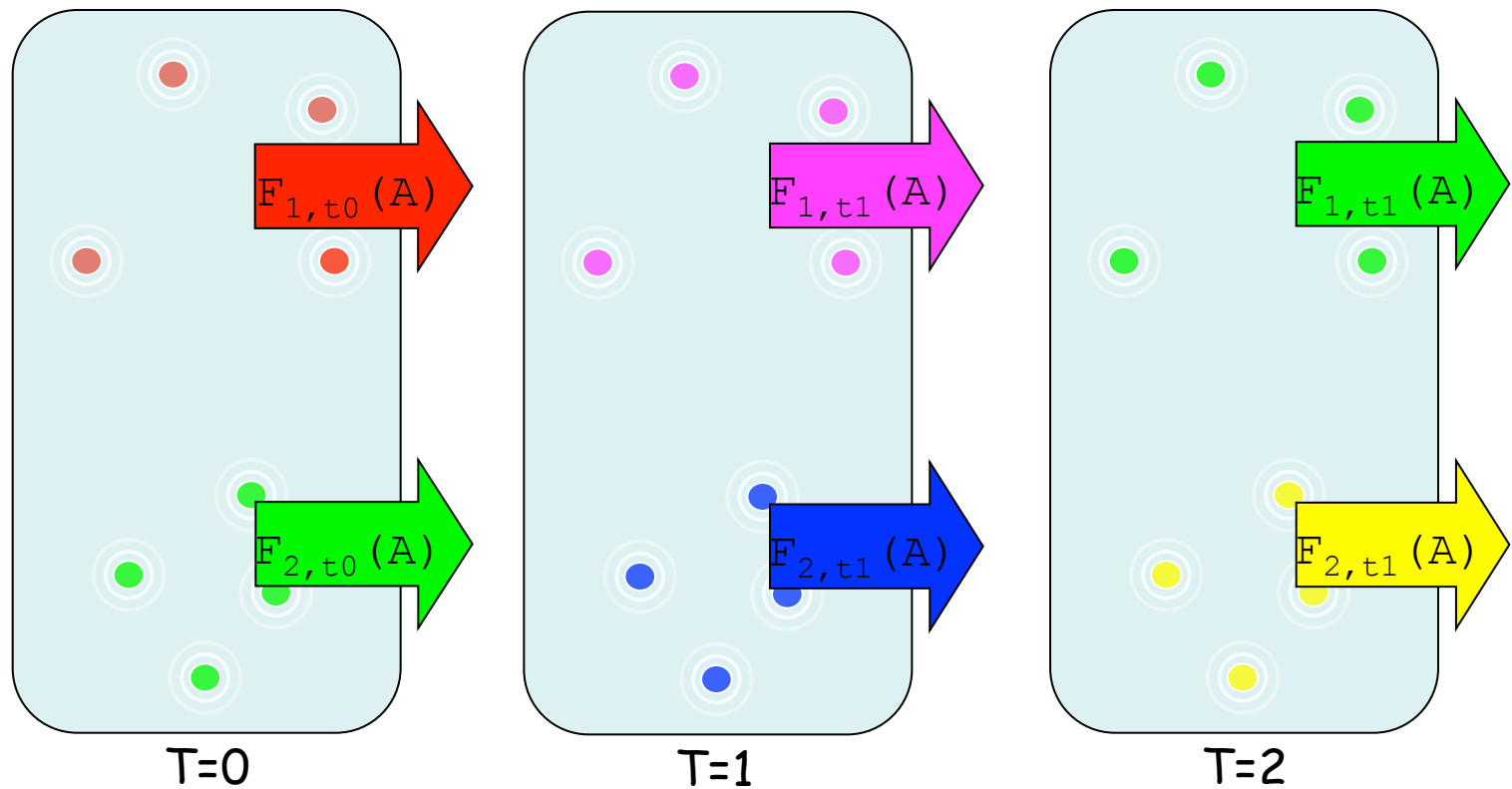
# DYNAMIC RECONFIGURATION

- ❑ TARGET PLATFORM: MICAZ
- ❑ EXTENSION OF THE THINK GENERIC COMPONENTS → VALENTINE OS



# TOWARDS SERVICE ORIENTED ARCHITECTURE

- FAST RECONFIGURATION ENABLES DYNAMIC AND ON-THE-FLY NEW SERVICES DEPLOYMENT



# COOPERATIVE INTELLIGENCE?

Combination of randomly and manually deployed sensors

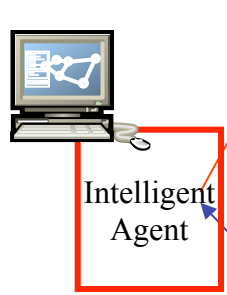
Authorised User



Activating

Sending

Sensor Net



Activating

Sending

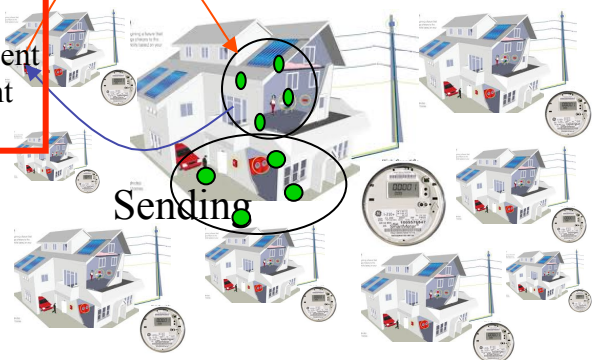


Activating

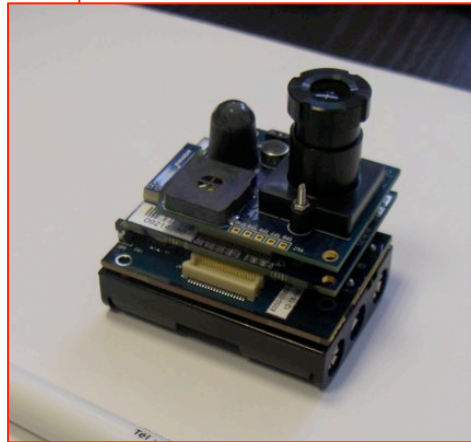
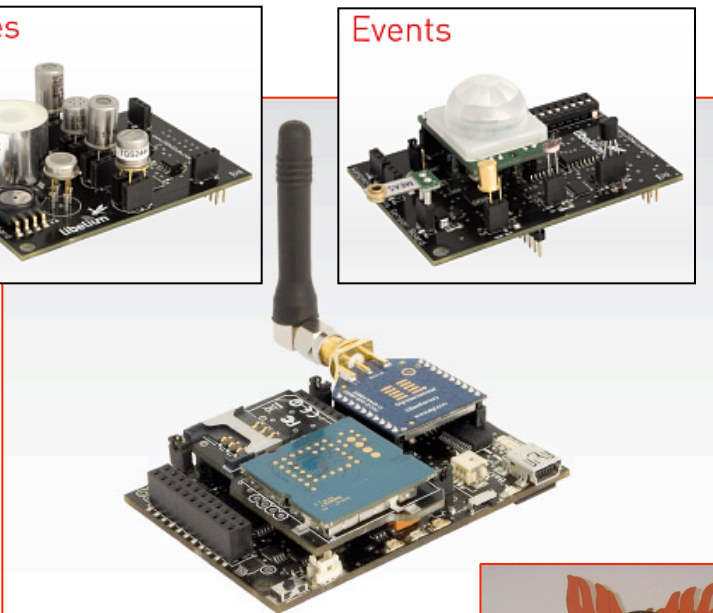
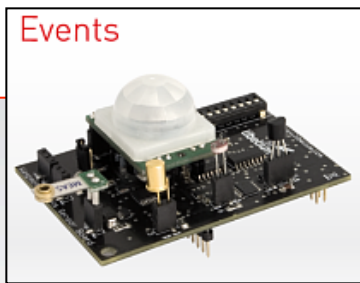


Intelligent Agent

Sending



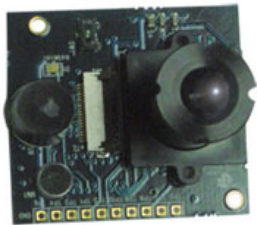
# CHALLENGING COOPERATION IMPLIES DIFFERENCES!



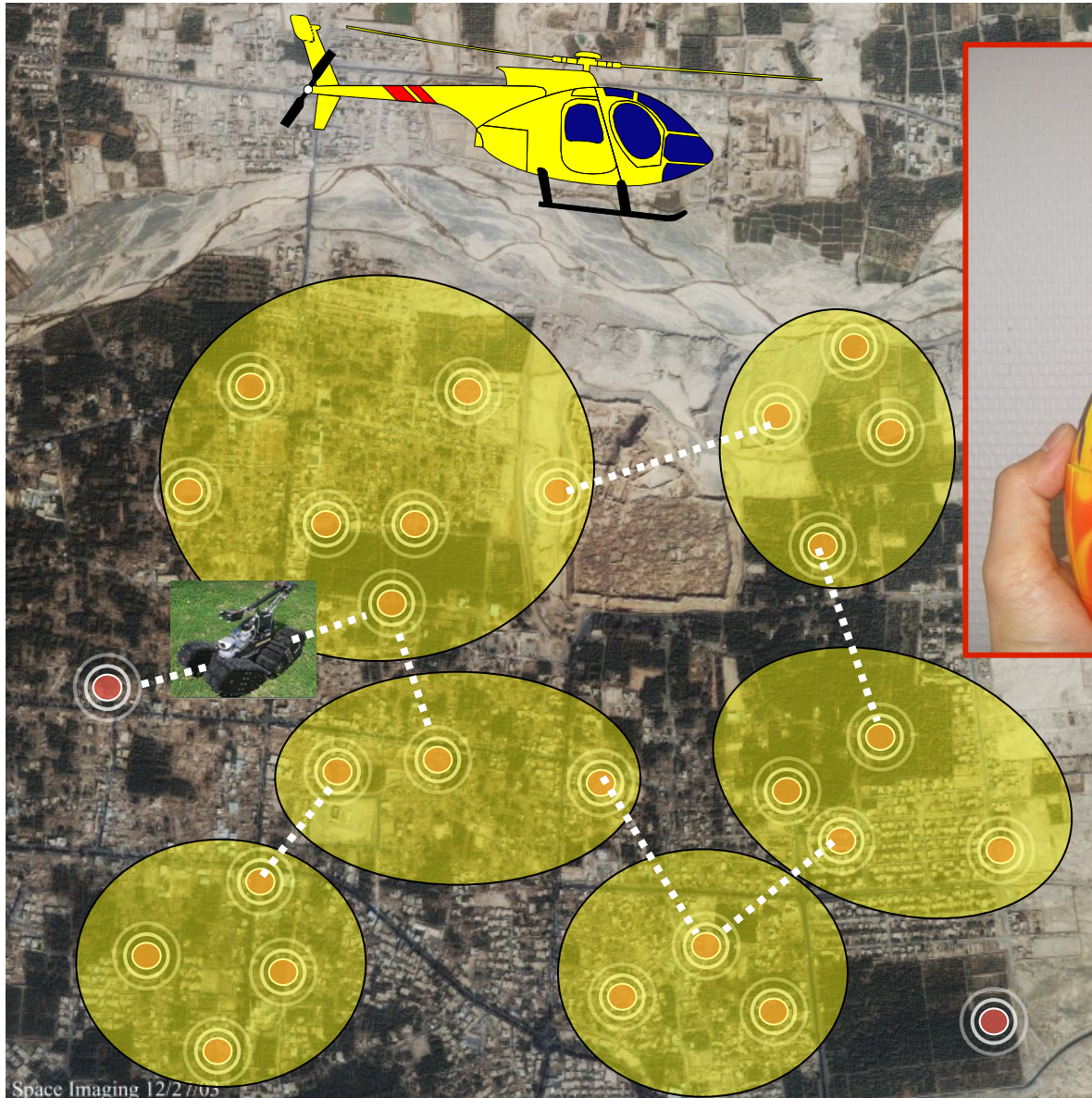
# ROBOT'S MOBILITY TO PRESERVE CONNECTIVITY



Imote2



Multimedia board





# SENSOR & ROBOTS SEARCH & RESCUE

- RESCUE COULD BE OPERATED IN SEVERAL PHASES (1)

Deploy in mass a WSN to get a first snapshot of the situation: images, radiation level, targets,...



# SENSOR & ROBOTS SEARCH & RESCUE

## □ RESCUE COULD BE OPERATED IN SEVERAL PHASES (2)

Based on collected data, optimize deployment/selection of autonomous robots



© Reuters

# SENSOR & ROBOTS SEARCH & RESCUE

## □ RESCUE COULD BE OPERATED IN SEVERAL PHASES (3)

Robots could serve as relay or install communication gateways to maintain WSN connectivity and increase data storage capability



# SENSOR & ROBOTS SEARCH & RESCUE

## □ RESCUE COULD BE OPERATED IN SEVERAL PHASES (4)

Sensor & Robots will continuously collaborate during the rescue process: localization, path optimization, remote sensing,...



# **CRITICALITY AND RISK- BASED SCHEDULING**

**BASIC APPROACH: PM2HW2N/ACM MSWIN 2009**

**CURRENT APPROACH: IEEE WCNC2010**

**WITH INTRUSION DETECTION RESULTS: IEEE RIVF2010**

**WITH RE-INFORCEMENT: IEEE ICDCN2011**

**JOURNAL PAPER IN JNCA, ELSEVIER**

# DON'T MISS IMPORTANT EVENTS!



WHOLE  
UNDERSTANDING  
OF THE SCENE IS  
WRONG!!!

WHAT IS CAPTURED

# HOW TO MEET SURVEILLANCE APP'S CRITICALITY

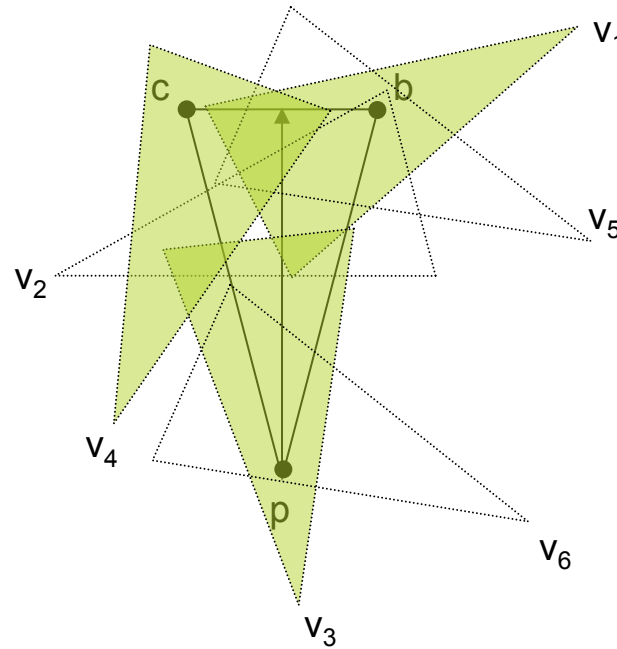
- ❑ CAPTURE SPEED CAN BE A « QUALITY » PARAMETER
- ❑ CAPTURE SPEED FOR NODE V SHOULD DEPEND ON THE APP'S CRITICALITY AND ON THE LEVEL OF REDUNDANCY FOR NODE V
- ❑ V'S CAPTURE SPEED CAN INCREASE WHEN AS V HAS MORE NODES COVERING ITS OWN FOV - COVER SET

# NODE'S COVER SET

$\text{Co}(V) = \{$   
 $\{V\},$   
 $\{V_1, V_3, V_4\},$   
 $\{V_2, V_3, V_4\},$   
 $\{V_3, V_4, V_5\},$   
 $\{V_1, V_4, V_6\},$   
 $\{V_2, V_4, V_6\},$   
 $\{V_4, V_5, V_6\}$   
 $\}$



$|\text{Co}(V)| = 7$





# CRITICALITY MODEL (1)

- LINK THE CAPTURE RATE TO THE SIZE OF THE COVER SET

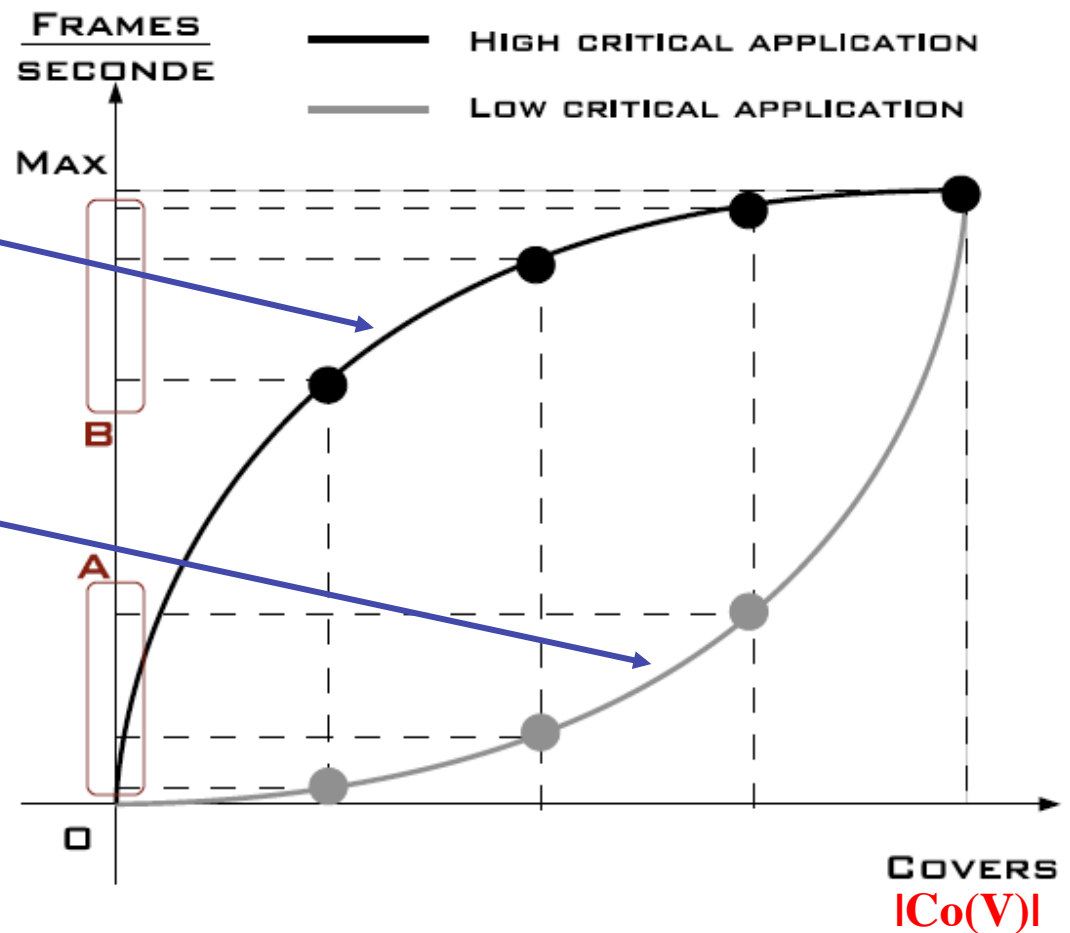
- HIGH CRITICALITY

- CONVEX SHAPE
- MOST PROJECTIONS OF X ARE CLOSE TO THE MAX CAPTURE SPEED

- LOW CRITICALITY

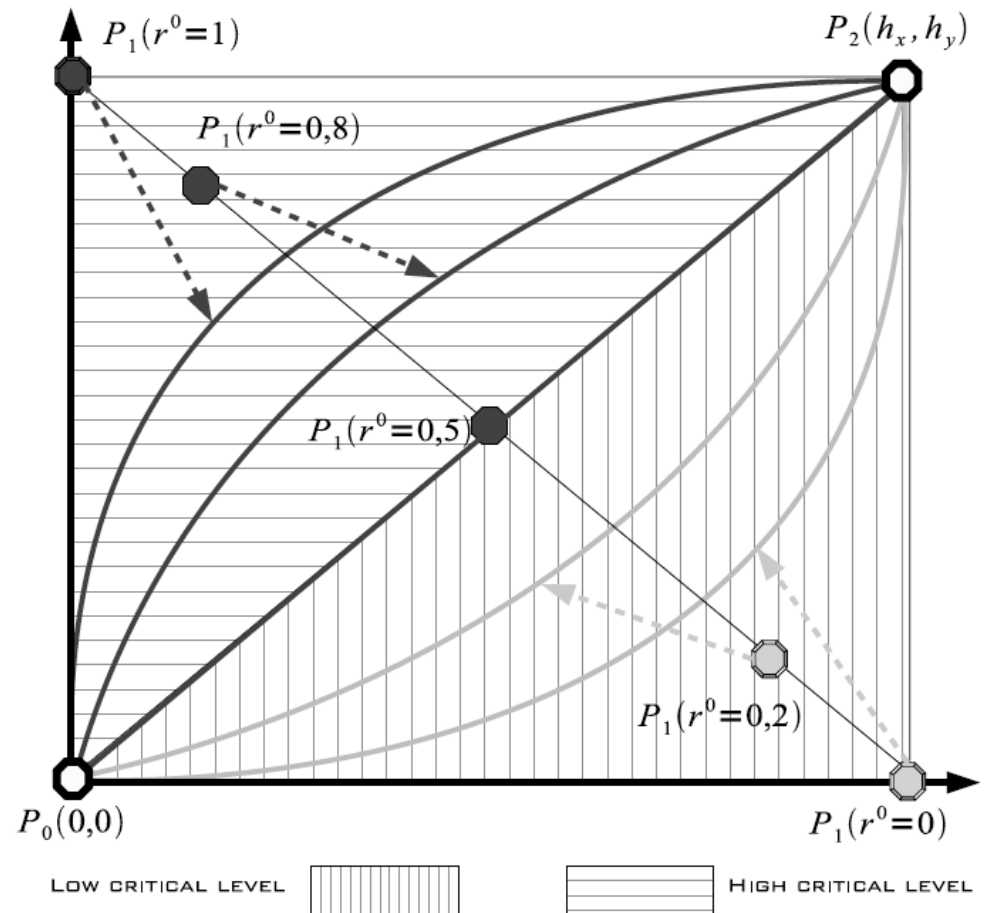
- CONCAVE SHAPE
- MOST PROJECTIONS OF X ARE CLOSE TO THE MIN CAPTURE SPEED

- CONCAVE AND CONVEX SHAPES AUTOMATICALLY DEFINE SENTRY NODES IN THE NETWORK



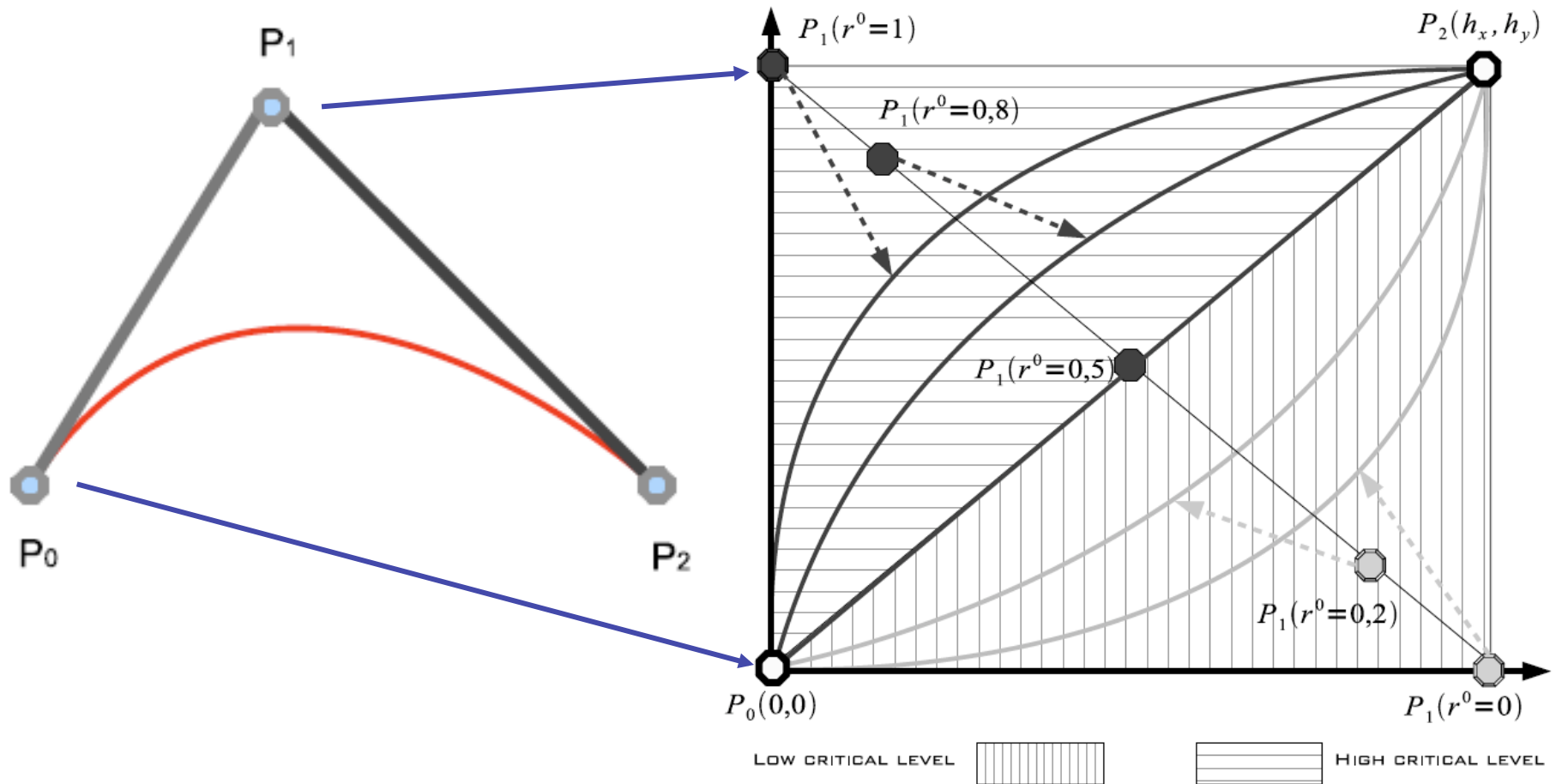
# CRITICALITY MODEL (2)

- ❑  $R^0$  CAN VARY IN  $[0,1]$
- ❑ BEHAVIOR FUNCTIONS (BV) DEFINES THE CAPTURE SPEED ACCORDING TO  $R^0$
- ❑  $R^0 < 0.5$ 
  - ❑ CONCAVE SHAPE BV
- ❑  $R^0 > 0.5$ 
  - ❑ CONVEX SHAPE BV
- ❑ WE PROPOSE TO USE BEZIER CURVES TO MODEL BV FUNCTIONS



# BEHAVIOR FUNCTION

$$B(t) = (1 - t)^2 * P_0 + 2t(1 - t) * P_1 + t^2 * P_2$$

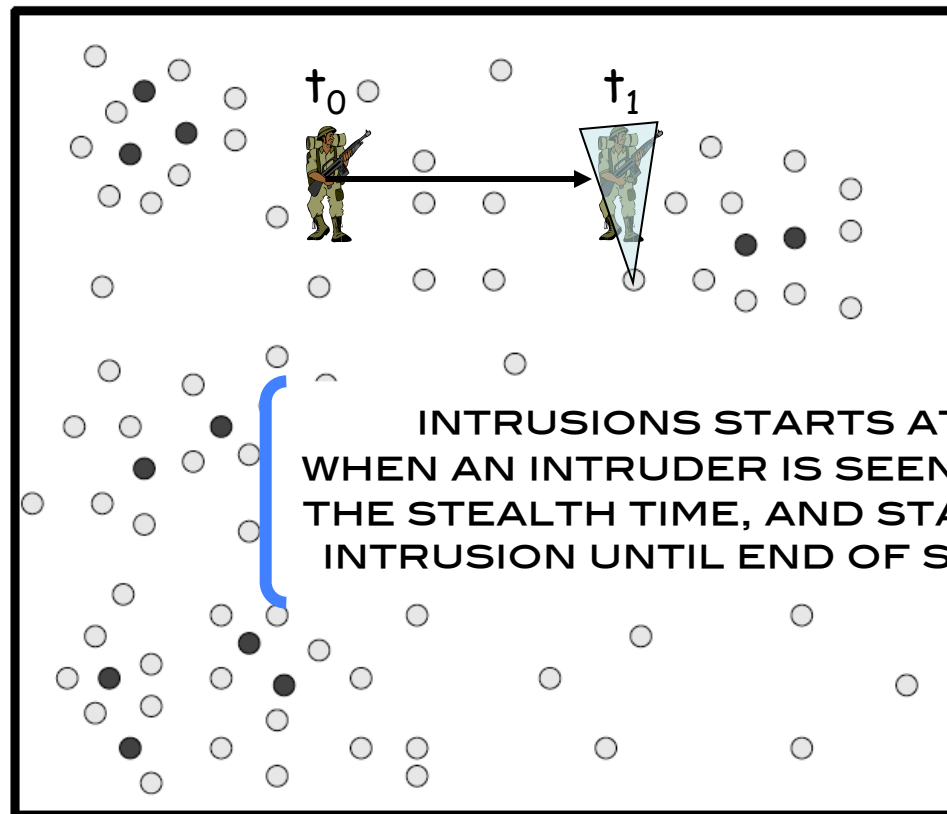


# RISK-BASED SCHEDULING

- ❑ **STATIC RISK-BASED SCHEDULING**
  - ❑  $R^{\circ} = \text{CTE}$  IN  $[0,1]$
- ❑ **DYNAMIC RISK-BASED SCHEDULING**
  - ❑ STARTS WITH A LOW VALUE FOR  $R^{\circ}$  (0.1)
  - ❑ ON INTRUSION, ALERT NEIGHBORHOOD AND INCREASES  $R^{\circ}$  TO A  $R_{\text{MAX}}$  VALUE (0.9)
  - ❑ STAYS AT  $R_{\text{MAX}}$  FOR  $T_A$  SECONDS BEFORE GOING BACK TO  $R^{\circ}$
- ❑ **DYNAMIC WITH REINFORCEMENT**
  - ❑ SAME AS DYNAMIC BUT SEVERAL ALERTS ARE NEEDED TO GET TO  $R^{\circ} = R_{\text{MAX}}$
  - ❑ GOING BACK TO  $R^{\circ}$  IS DONE IN ONE STEP

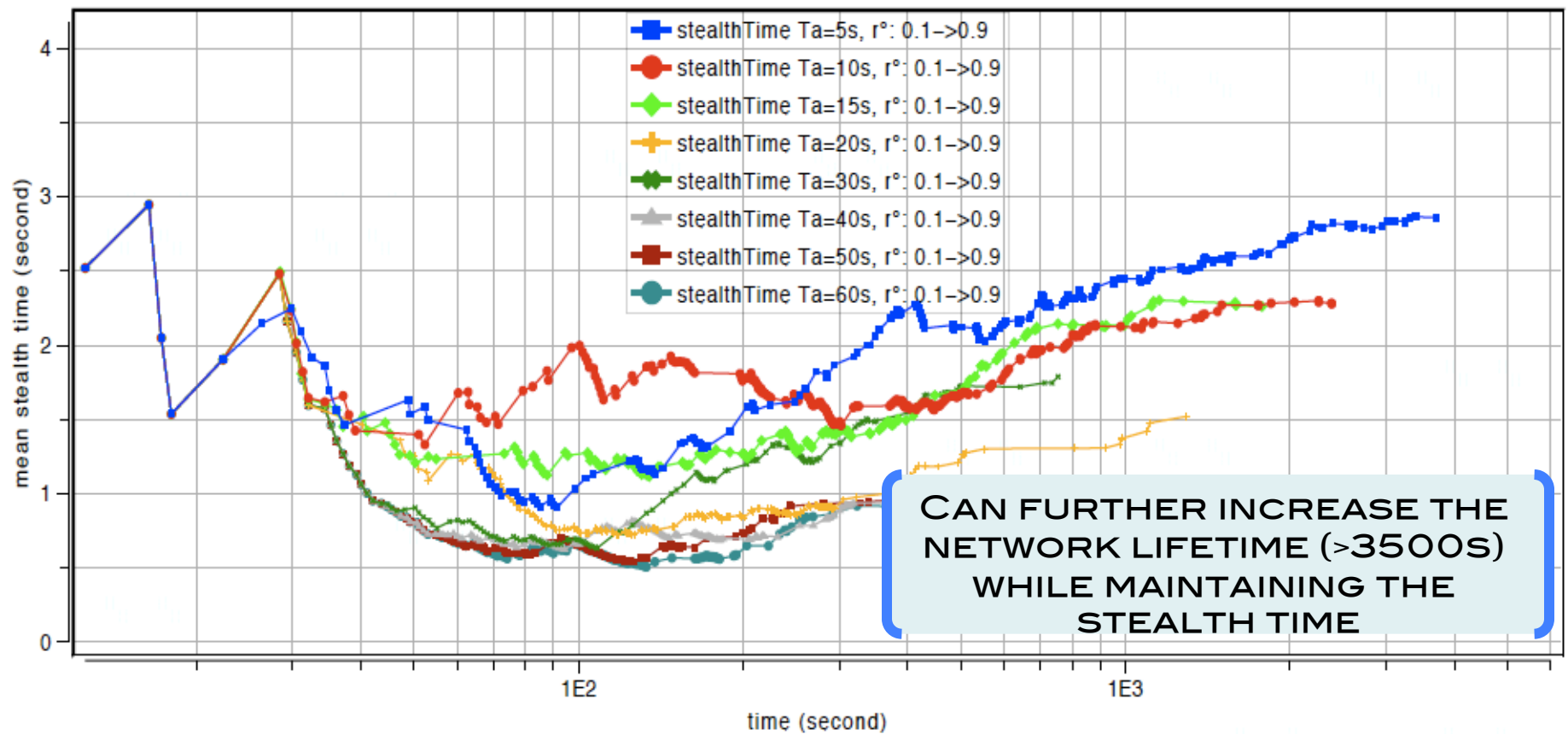
# MEAN STEALTH TIME

$T_1 - T_0$  IS THE INTRUDER'S  
STEALTH TIME  
VELOCITY IS SET TO 5M/S



# DYNAMIC SCHEDULING

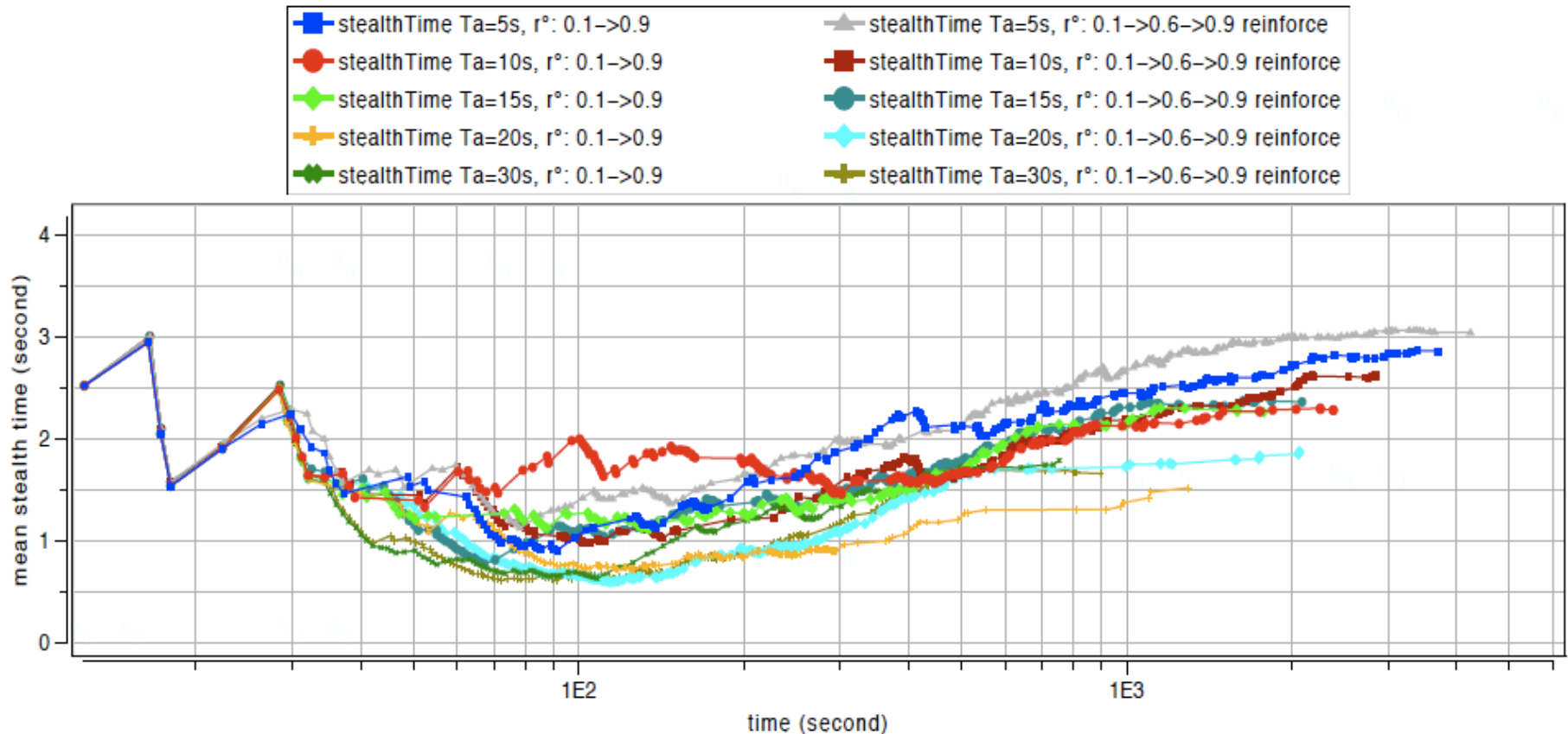
□  $R^0=0.1$ ,  $R_{MAX}=0.9$ ,  $T_A=5,10,15,20..60s$



# DYNAMIC WITH REINFORCEMENT (1)

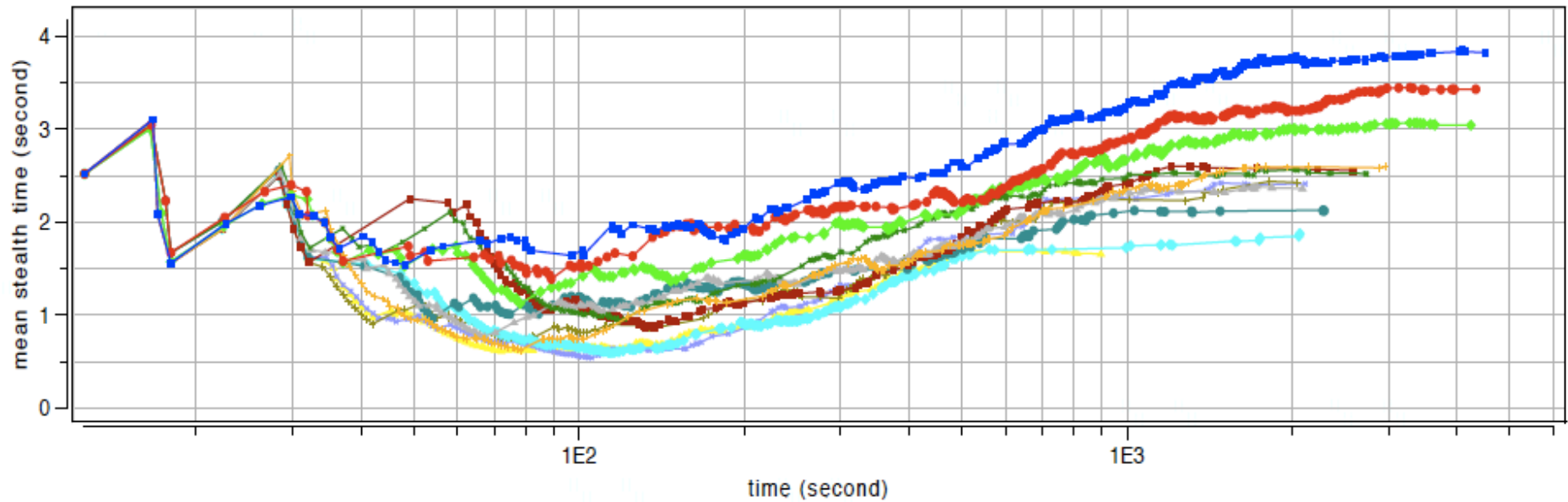
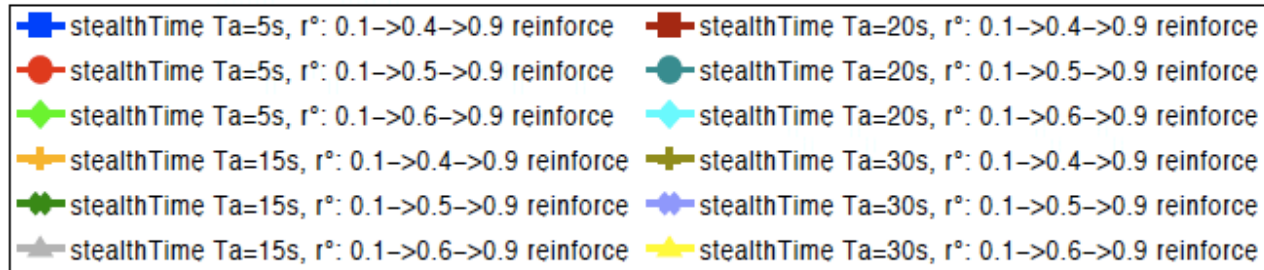
□  $R^0 = 0.1 \rightarrow I_R = 0.6 \rightarrow R_{MAX} = 0.9$

□ 2 ALERT MSG TO HAVE  $I_R = I_R + 0.1$



# DYNAMIC WITH REINFORCEMENT (2)

- $R^o = 0.1 \rightarrow I_R = 0.4/0.5/0.6 \rightarrow R_{MAX} = 0.9$
- 2 ALERT MSG TO HAVE  $I_R = I_R + 0.1$

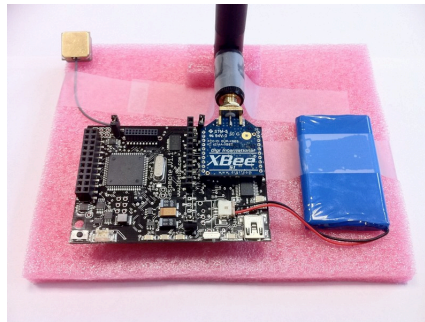
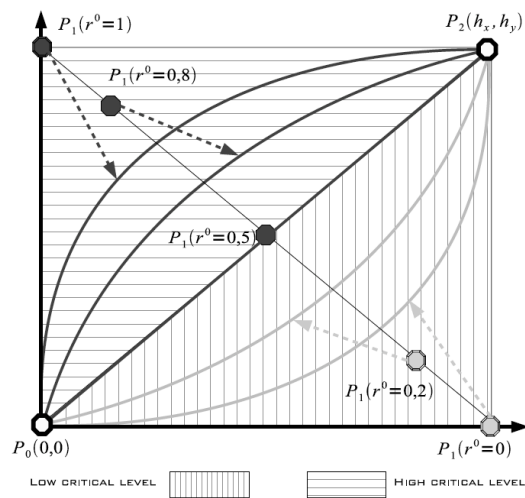




# SENSORS & ROBOTS

## PROPOSE NEW INTERACTION SCHEMES

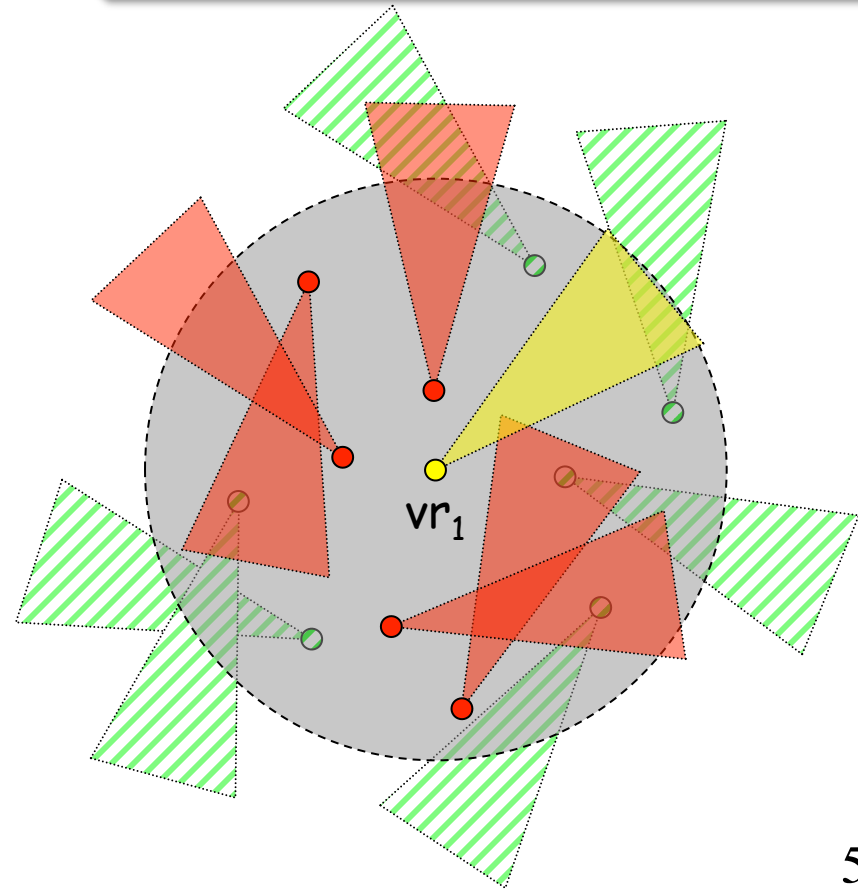
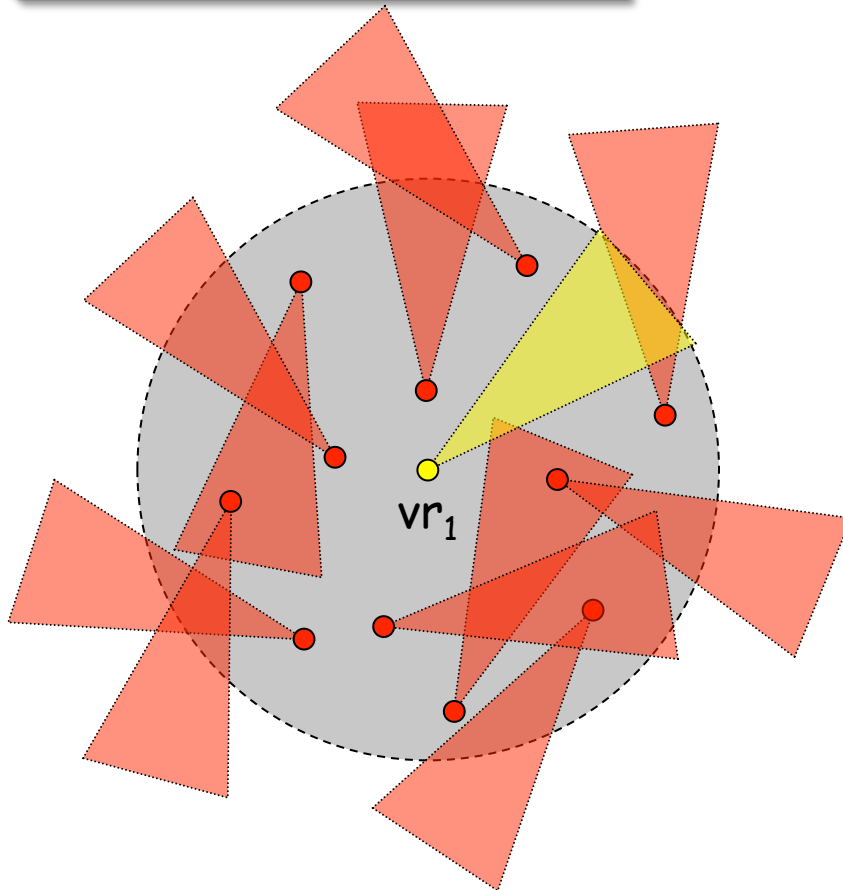
- ❑ USE THE CRITICALITY MODEL TO CONTROL BOTH SENSORS AND ROBOTS
- ❑ PROTOTYPING ON REAL HARDWARE, COLLABORATION WITH U. KYOTO, JAPAN



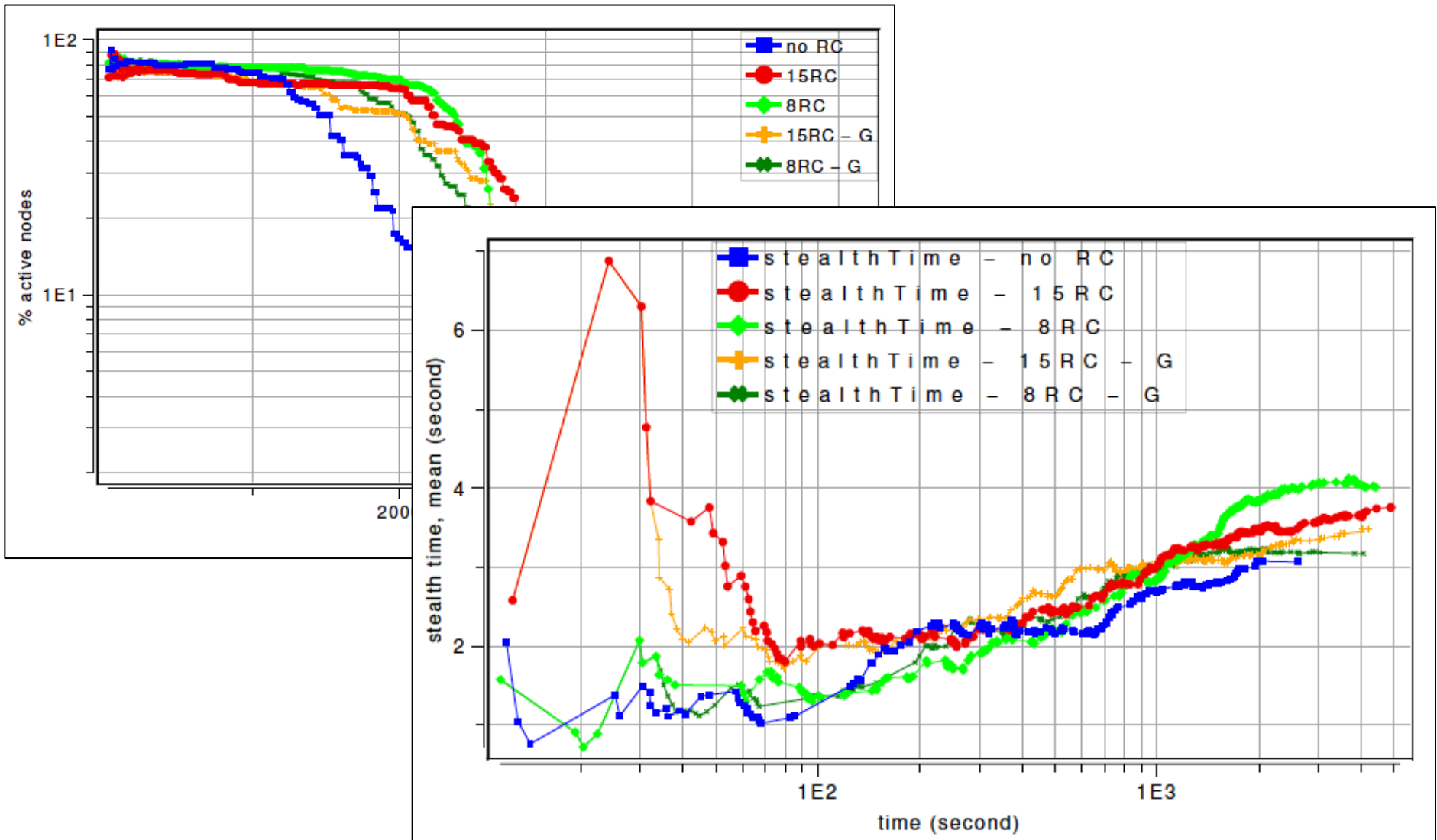
# COOPERATION WITH CAMERAS ON MOBILE ROBOTS

Fixed image sensors near a mobile camera can decrease their criticality level

**ONLY** fixed image sensors whose FoV's center is covered by a mobile camera **CAN** decrease their criticality level



# IMPACT ON LIFETIME & STEALTH TIME



# CONCLUSIONS

- ❑ SENSOR NETWORKS CAN PROVIDE LARGE SCALE AWARENESS TO SETUP THE FOUNDATION FOR **AMBIENT INTELLIGENCE**
- ❑ SENSORS & ROBOTS INTERACTIONS ARE CHALLENGING BUT HAVE VERY PROMISING **COOPERATIVE** ISSUES
- ❑ SURVEILLANCE APPLICATIONS HAVE CRITICALITY ISSUES THAT **MUST** BE TAKEN INTO ACCOUNT IN THE DESIGN PROCESS
- ❑ **NEED** TOOLS TO CAREFULLY STUDY THE ENTIRE SYSTEM

# MULTIDISCIPLINARY RESEARCH

EVALUATION AND SIMULATION

DISTRIBUTED  
SYSTEMS

NETWORKING

APPLICATIONS

ALGORITHMS

HARDWARE

ENERGY  
CONSIDERATIONS

NETWORK

SIGNAL  
IMAGE/VIDEO  
PROCESSING

OS  
MIDDLEWARE  
SOFT. ENG.

DATA MNGT

HARDWARE  
RADIO

[ MIDDLEWARE/APP. ]  
ISSUES WE  
ADDRESS

SENSOR'S OS

CBSE for SENSOR NODE  
DYNAMIC  
RECONFIGURATION

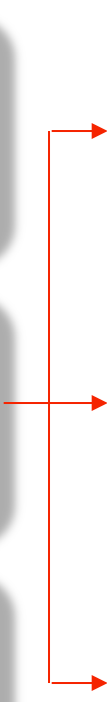
SUPERVISION  
PLATFORM

SERVICE-ORIENTED  
SERVICE REPOSITORY

APPLICATIONS

ADAPTIVE APPLICATION

QOS



ENERGY  
CONSIDERATIONS

NETWORK

SIGNAL  
IMAGE/VIDEO  
PROCESSING

OS  
MIDDLEWARE  
SOFT. ENG.

DATA MNGT

HARDWARE  
RADIO

# NETWORK ISSUES WE ADDRESS

ORGANIZATION  
OVERLAYS

VIDEO COVERAGE  
SELECTION &  
WAKE-UP MECHANISM

TRANSPORT

LOAD-REPARTITION  
CONGESTION CONTROL

ROUTING

MULTI-PATHS  
ROUTING  
OPTIMIZED  
GEOGRAPHIC ROUTING

MAC  
RESOURCES  
ALLOCATION

« URGENT » MAC LAYER  
CRITICALITY-AWARE  
MAC LAYER

QoS

# IMAGE SENSOR SIMULATION MODEL UNDER OMNET++

- ❑ COMMUNICATION LAYERS ARE VERY IMPORTANT FOR WSN
- ❑ USE SPECIFIC SIMULATOR

The image displays two screenshots from the OMNeT++ simulation environment. The left screenshot shows the internal structure of a node (SN.node[0]), highlighting the communication layer components: Radio, MAC, and Routing. The right screenshot shows a network topology with multiple nodes (node0 to node59) and a console window displaying simulation logs. An orange callout box is overlaid on the right screenshot, containing the text: "Need to know the power consumption for capturing an image, processing/compressing an image & transmitting an image...". A small inset image in the bottom left corner shows a physical image sensor module.



# STUDY THE IMPACT OF COMMUNICATION LAYER ON SURVEILLANCE QUALITY

The image displays a simulation environment for a network-based surveillance system. It consists of three main components:

- Top Left:** A small window titled "79(33.8) <-46(1)" showing a real-time video feed of a desert landscape with a road and some vegetation.
- Center:** A network map window titled "(SN) SN" showing a complex network of nodes (represented by colored circles) and links. The map includes a toolbar with "RUN", "STOP", and "Zoom: 0.79x".
- Right:** A terminal window titled "OMNeT++/Tkenv - SN" showing the simulation's output. It displays statistics such as "Msgs created: 667040", "Msgs present: 1867", and "Ev/simsec: 15059.8". The terminal output shows a series of messages, including "Timer message", "capture...", and "Application Sending [image] of size 288 bytes to communication layer". It also shows intrusion detection events: "Node 96: INTRUSION SEEN", "Node 148: INTRUSION SEEN", and "Node 5: INTRUSION SEEN".

```
SN.node[46].Application Sending [image] of size 288 bytes to communication layer
SN.node[46].Application Sending [image] of size 288 bytes to communication layer
SN.node[46].Application Sending [image] of size 288 bytes to communication layer
SN.node[46].Application Sending [image] of size 288 bytes to communication layer
SN.node[46].Application Sending [image] of size 288 bytes to communication layer
SN.node[46].Application Sending [image] of size 288 bytes to communication layer
SN.node[46].Application Sending [image] of size 288 bytes to communication layer
SN.node[46].Application Sending [image] of size 288 bytes to communication layer
SN.node[46].Application Sending [image] of size 288 bytes to communication layer
SN.node[46].Application Sending [image] of size 288 bytes to communication layer
SN.node[46].Application Sending [image] of size 288 bytes to communication layer
SN.node[46].Application Sending [image] of size 288 bytes to communication layer
SN.node[46].Application Sending [image] of size 288 bytes to communication layer
SN.node[46].Application Sending [image] of size 288 bytes to communication layer
SN.node[46].Application Node 46 -> REAL IMAGE(1) to node 79
SN.node[96].Application Node 96: INTRUSION SEEN

SN.node[148].Application Node 148: INTRUSION SEEN

SN.node[148].Application Sending [alert] of size 30 bytes to communication layer
SN.node[148].Application Node 148: INTRUSION SEEN
pert.coverage 99.8628%
nb active nodes 100%

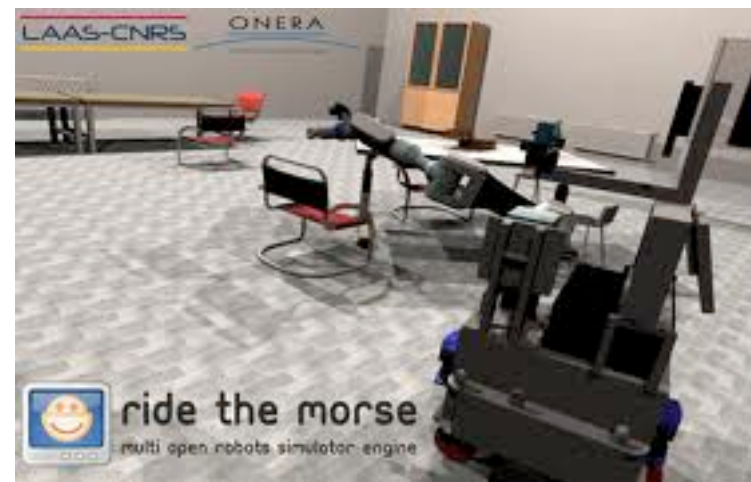
SN.node[5].Application Node 5: INTRUSION SEEN

SN.node[5].Application Sending [alert] of size 30 bytes to communication layer
SN.node[6].Application Node 6: INTRUSION SEEN
SN.node[6].Application Sending [alert] of size 30 bytes to communication layer
SN.node[5].Application Node 5: INTRUSION SEEN
SN.node[6].Application Node 6: INTRUSION SEEN
SN.node[124].Application Node 124: INTRUSION SEEN
SN.node[124].Application Sending [alert] of size 30 bytes to communication layer
SN.node[5].Application Node 5: INTRUSION SEEN
SN.node[24].Application Node 24: INTRUSION SEEN
SN.node[24].Application Sending [alert] of size 30 bytes to communication layer
SN.node[6].Application Node 6: INTRUSION SEEN

SN.node[79].Application Node 79: WRITES IMAGE FILE(1) from node 10
SN.node[79].Application Node 79: DISPLAY REAL IMAGE(1) from node 10
```

# ROBOT SIMULATORS

- ❑ MOBILITY, EXPLORATION, NAVIGATION, TRACKING, CONTROL AND DESIGN ARE VERY IMPORTANT FOR ROBOTS
- ❑ USE SPECIFIC ROBOT SIMULATORS



# SENSORS & ROBOTS ENABLE REALISTIC INTERACTION STUDIES

Sensor specific simulator for communication stack

Get robot's position from robot simulator

Re-use fine-grained communication protocols and complex radio models

Re-use complex hardware (laser scan, ...) and control software (navigation stacks,...)

The image displays two main components. On the left is a network diagram window titled '(SN) SN' showing a central 'SN' node connected to a 'coordinator' and 'intrusion' nodes. Below these are numerous 'node' icons (e.g., node[0], node[1], node[2], node[3], node[4], node[5], node[7], node[9], node[12], node[13], node[14], node[15], node[17], node[20], node[21], node[22], node[23], node[25], node[27], node[28], node[29]) arranged in a grid. On the right is a 3D robot simulator window titled 'Blender' showing a red robot on a grassy hillside. A console window on the right side of the Blender window displays text: 'Multi Open Robots Simulation' and 'Display Help: Press the H key'. Red arrows point from the network diagram to the 3D simulator, indicating data flow. A blue callout box on the left contains the text 'Re-use fine-grained communication protocols and complex radio models'. A green callout box on the right contains the text 'Re-use complex hardware (laser scan, ...) and control software (navigation stacks,...)'. The text 'Get robot's position from robot simulator' is positioned above the 3D simulator window.

# THANK YOU!



PROF. CONGDUC PHAM  
[HTTP://WWW.UNIV-PAU.FR/~CPHAM](http://www.univ-pau.fr/~cpham)  
UNIVERSITÉ DE PAU, FRANCE

