Assistive technologies

Stefano Chessa

Dipartimento di Informatica, Università di Pisa

Lezione TFA, 21 Maggio 2015

Background

Step counter

Hearth rate

Sensors in smartphones

Fall detector

Blood pressure

Trend in sensors for assistive technologies

- Devices embedding sensors and artificial intelligence
  - To analyze locally sensed data in a “intelligent” way
  - Can train the devices to recognize specific situations, movements etc.
- Advantages
  - Devices even more efficient and smaller
  - Even more pervasive…
  - … and interoperable at high level with other users devices

Intelligent sensors in assistive technologies

- Provide solutions to recognize the activities performed by elderly or disabled
- By means of simple sensors
  - Wearable: step counters, hearth rate, accelerometers on arms/legs,…
  - Environmental: PIRs, localization, door switches, sensorized carpets,…

Intelligent sensors in assistive technologies

However, the requests for activity recognition can very demanding

- Recognize even complex user activities:
  - Relaxing
  - Exercising
  - Cooking
  - Socializing
  - …

The challenge…

Recent projects

SEVENTH FRAMEWORK PROGRAMME

doremi

- “Decrease of cOgnitive decline, malnutRition and sedEntariness by elderly empowerment in lifestyle Management and social Inclusion”
- Novembre 2013 – Ottobre 2016.
- Objective ICT-2013.5.1 «personalized health, active aging and independent living»

rubicon

- “Robotic UBIquitous COgnitive Network”
- Aprile 2011 – Marzo 2014.
- Obiettivo FP7-ICT-2009-6 “robotics and cognitive systems”.

Summary of the lecture

- Presentation of RUBICON and DOREMI
- Some HW platforms
- Arduino
- An example

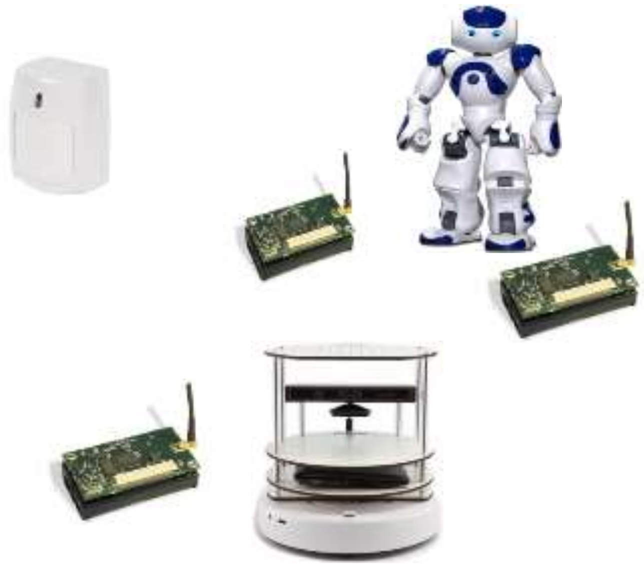
rubicon

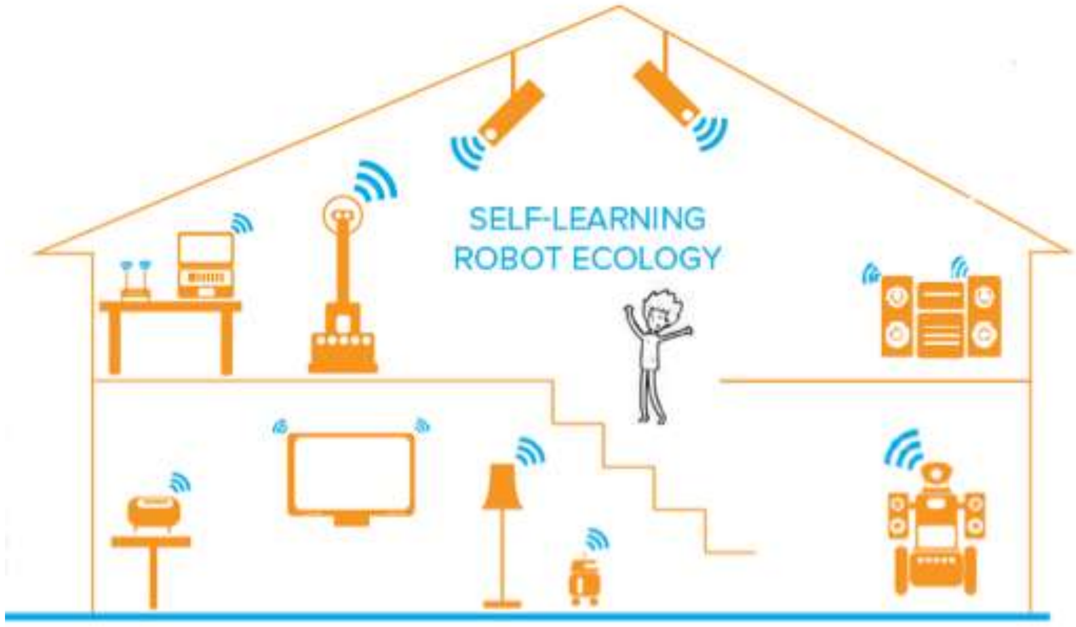
Robotic UBIquitous COgnitive Network

EU FP7 - www.fp7rubicon.eu

RUBICON Goal

Develop self-learning robotic ecologies





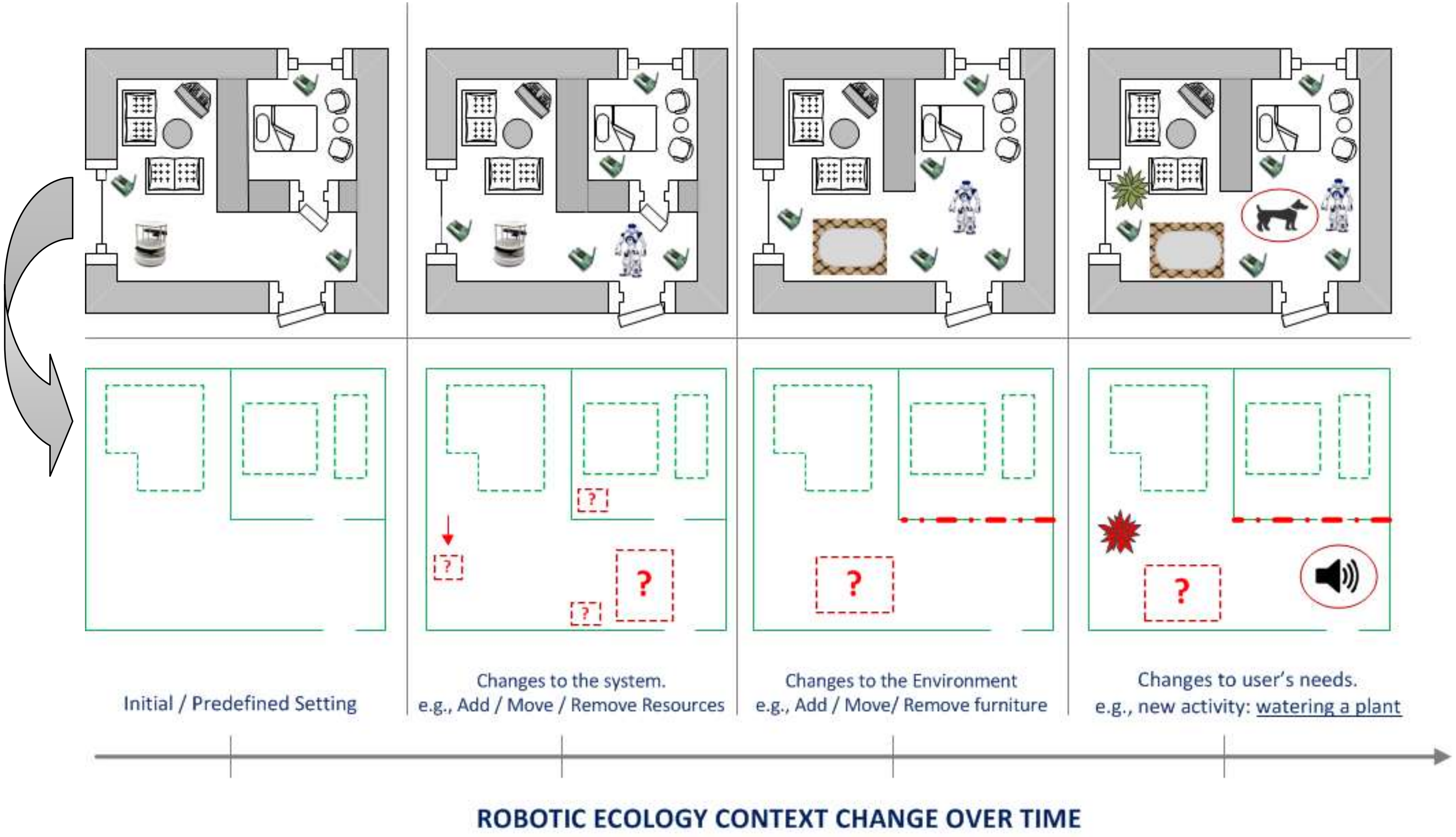
Problem Addressed

Current Robotic Ecologies suffer of brittle behaviour and lack the ability to proactively and smoothly adapt to changing and evolving situations :

- Difficulty in interpreting noisy and uncertain sensors
- Excessive Reliance on Symbolic Representations
- Excessive Reliance on Humans

=> Solutions are still difficult and prohibitively costly to deploy and maintain in real world applications !

Need for learning solution



Initial / Predefined Setting

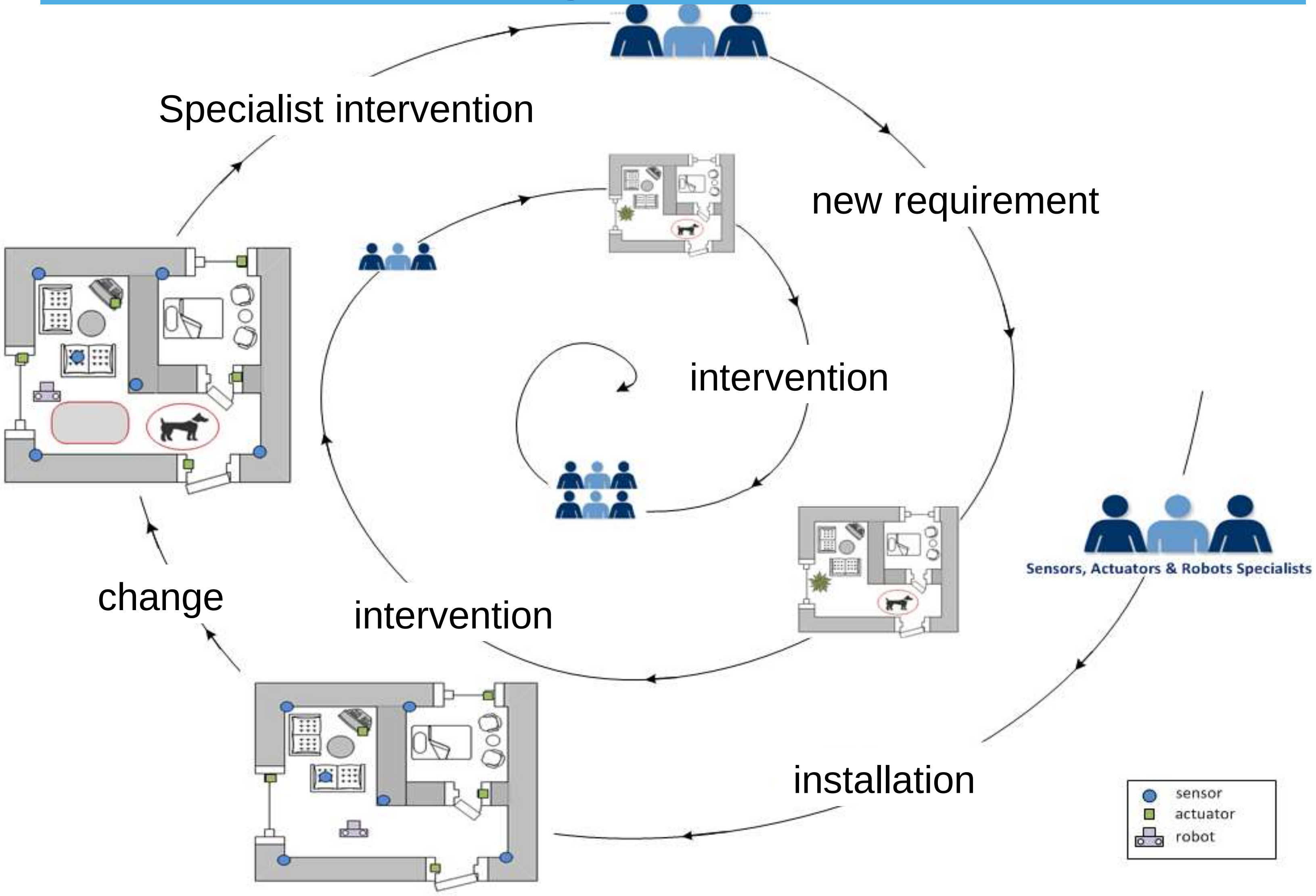
Changes to the system.
e.g., Add / Move / Remove Resources

Changes to the Environment
e.g., Add / Move/ Remove furniture

Changes to user's needs.
e.g., new activity: watering a plant

ROBOTIC ECOLOGY CONTEXT CHANGE OVER TIME

Need for learning solution



Specialist intervention

new requirement

intervention

change

intervention

installation

Self-Sustaining Learning - Impact

Reduce

need for

Programming

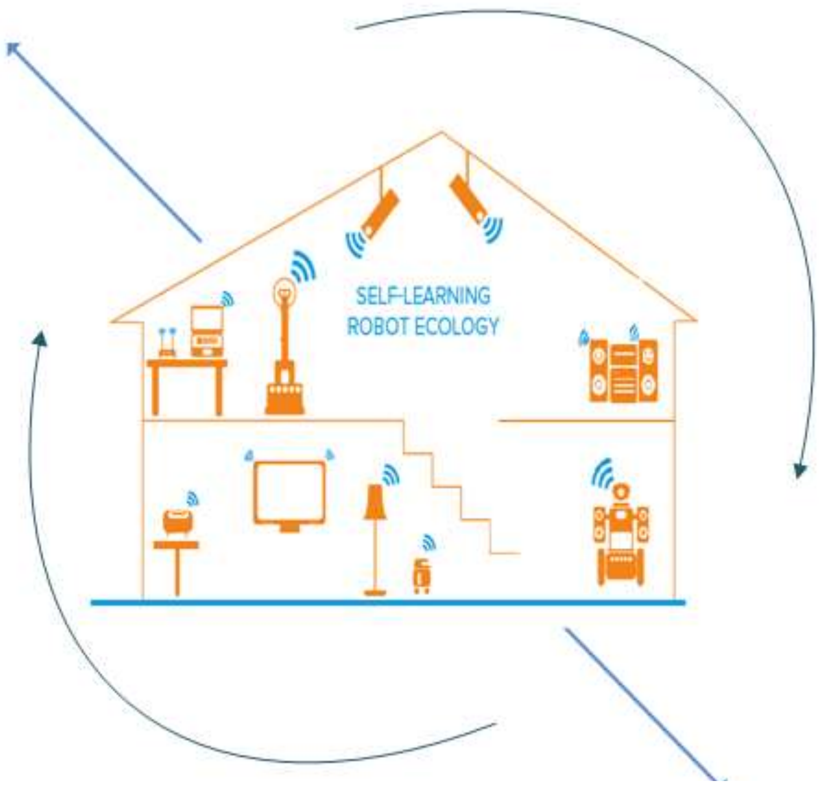
Configure

Train

Supervise

Robotic ecology

solutions



RUBICON delivers efficient self-learning solution for robotic ecologies

Increase

Adaptability

Flexibility

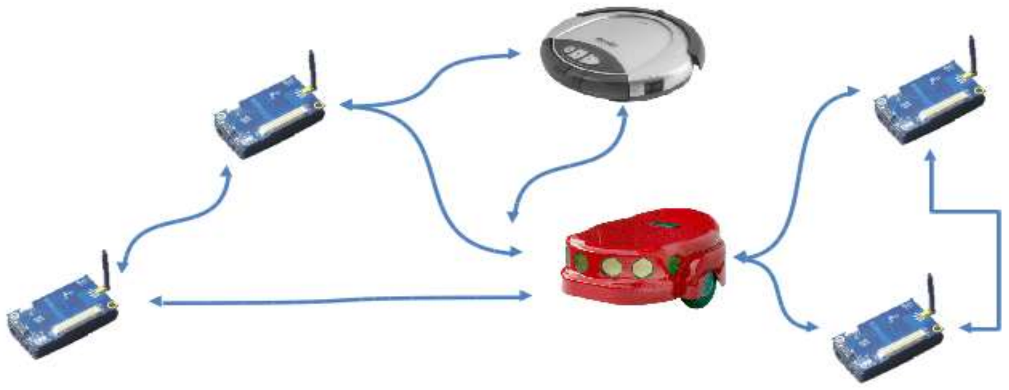
Robustness

Fault tolerance

Open new

application areas

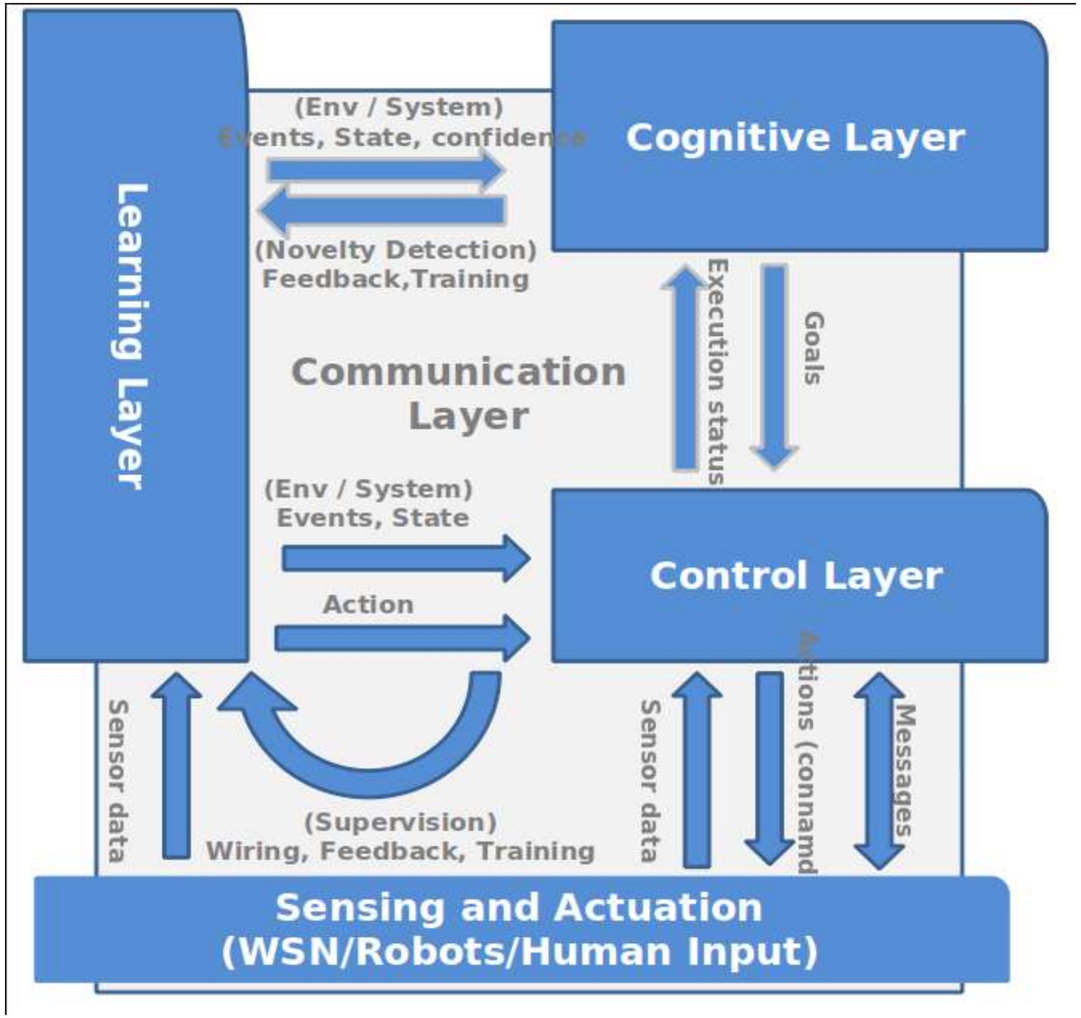
A Robotic Ecology Solution



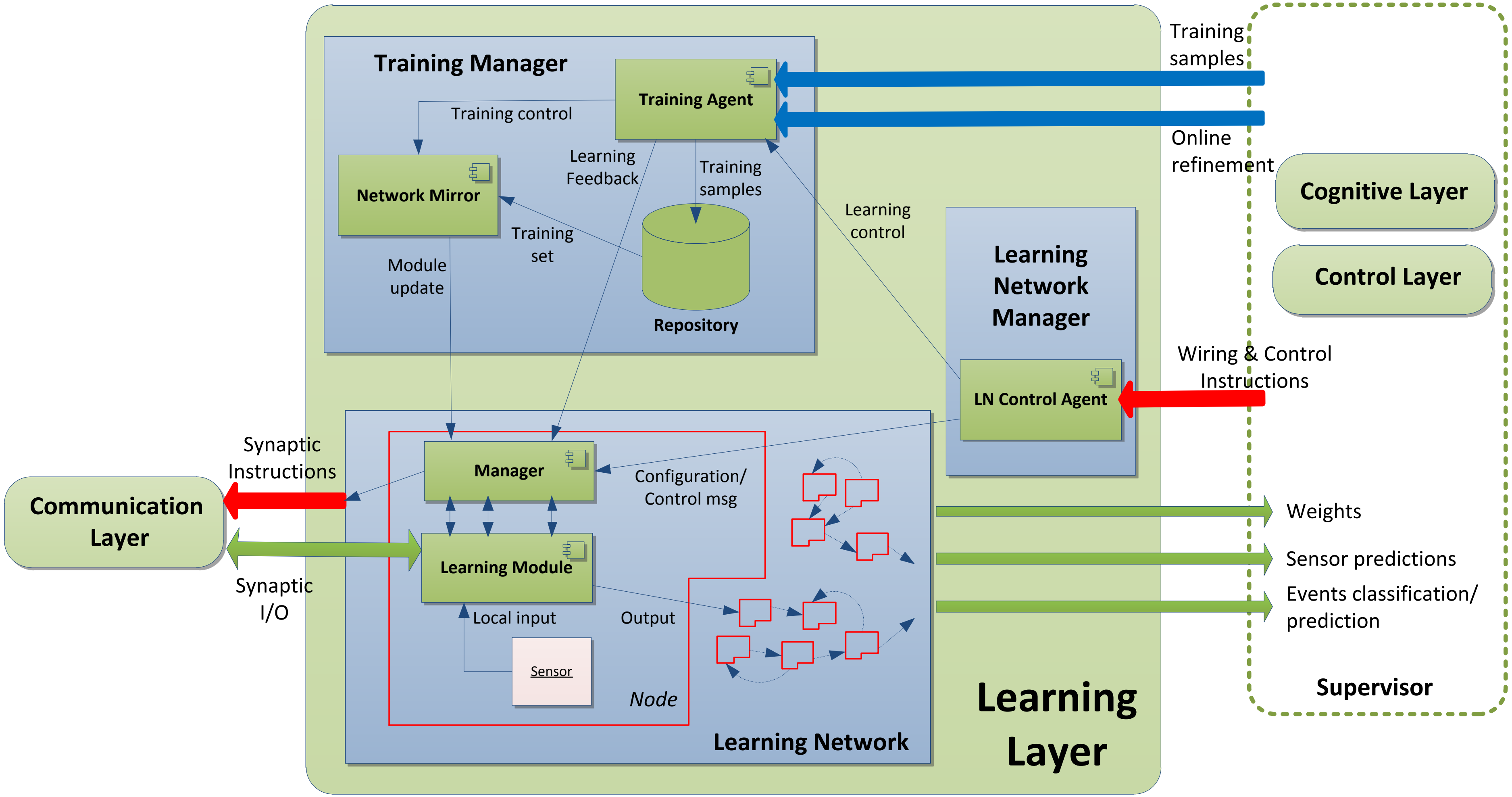
The nodes of a RUBICON ecology mutually support one another's learning:

- cooperate in using past experience to improve performance and adjust to changing situations
- shared, open and distributed learning infrastructure
- mutually self-sustaining system

RUBICON approach



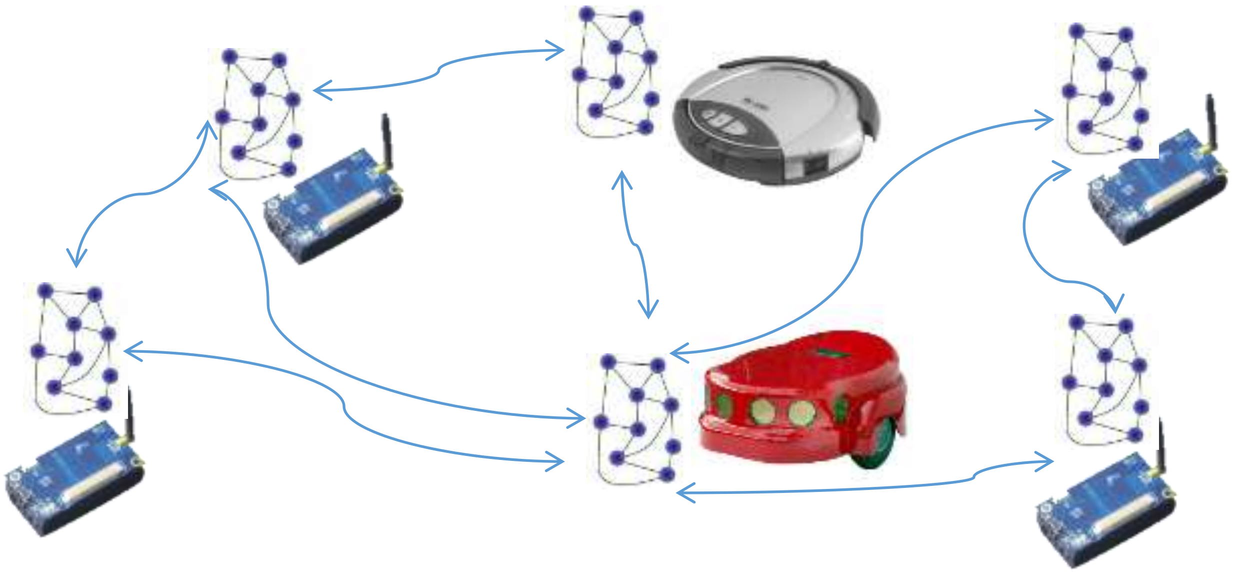
Learning Layer Architecture



Learning Layer Subsystems

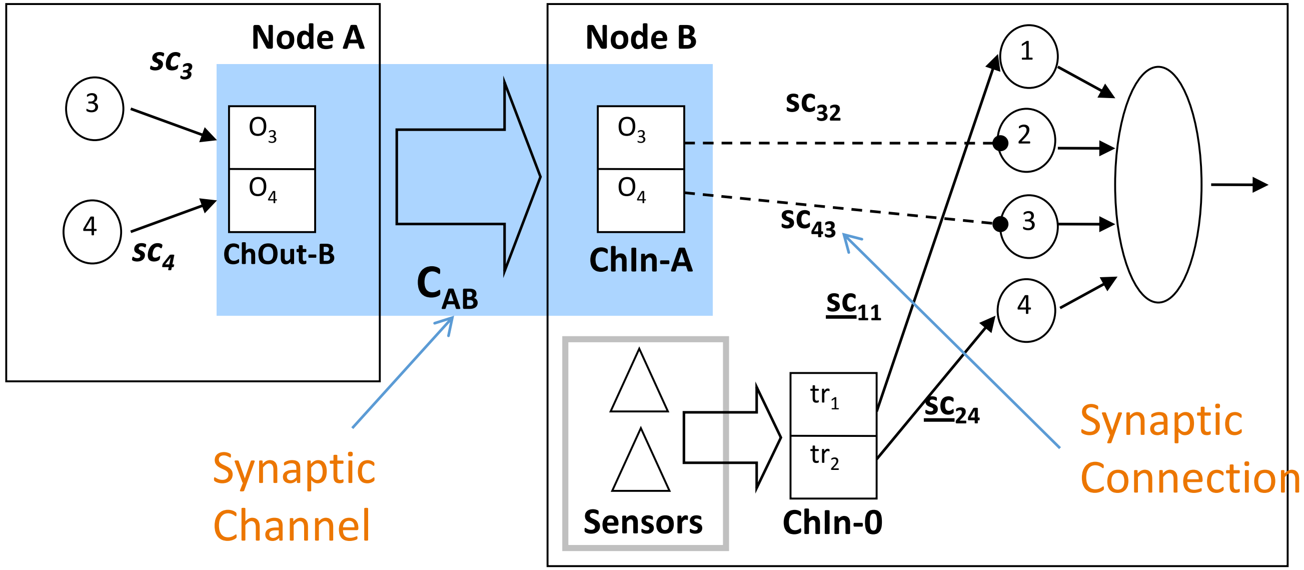
- Learning Network (LN)
  - Implements a distributed neural computation to produce the Learning Layer predictions
  - Embedded Echo State Networks (ESN)
- Learning Network Manager (LNM)
  - Configuration and control of the Learning Layer
  - Interface to higher RUBICON layers
- Training Manager (TM)
  - Manages the learning processes of the LN
  - Ensures LN reconfigurability

Distributed Neural Computation



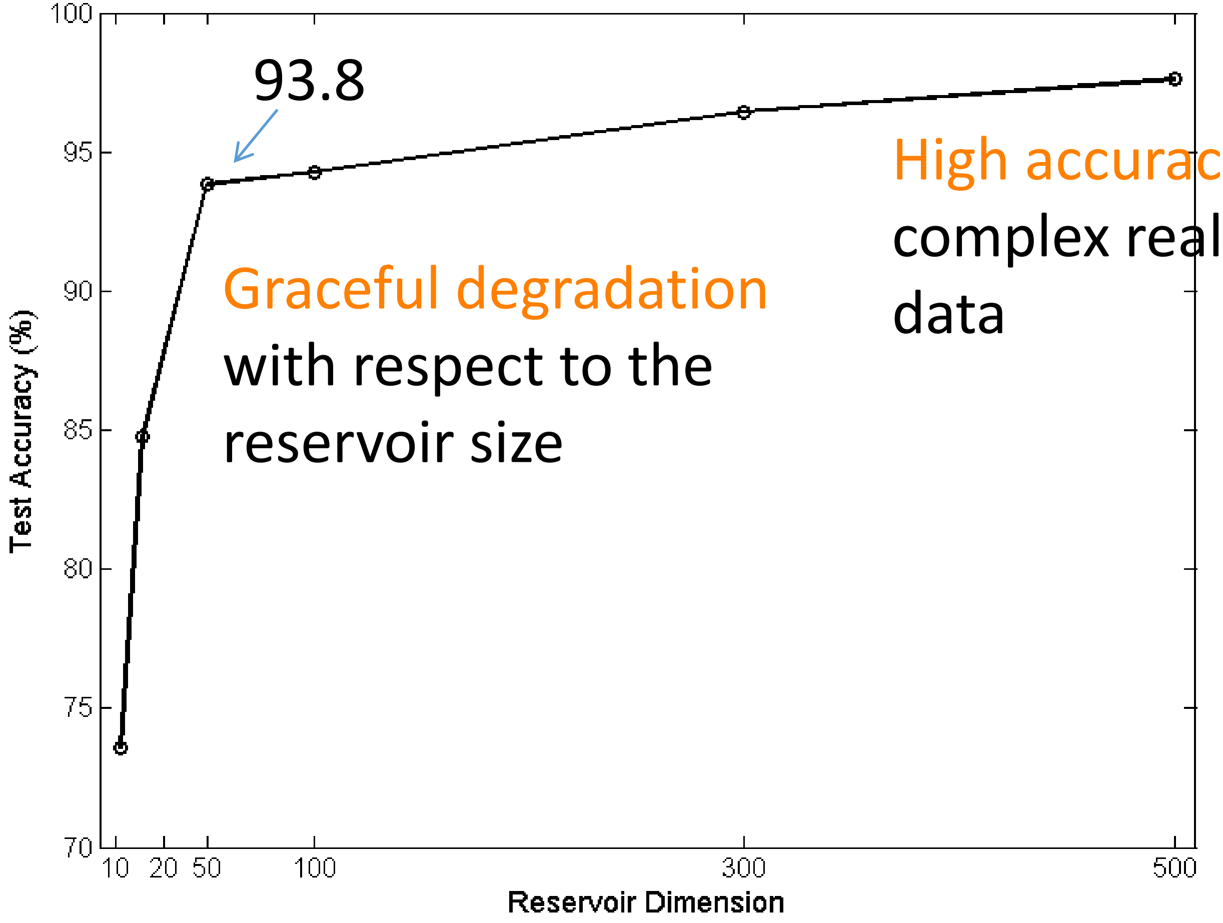
- Embed learning to implement an ecology memory distributed in the environment
- Echo State Network as a parsimonious recurrent neural model capable of processing complex time-dependent data

Synaptic Communication



- Deliver local and remote sensor/neural data to the input neurons
  - Information demultiplexing
  - Quality of Service

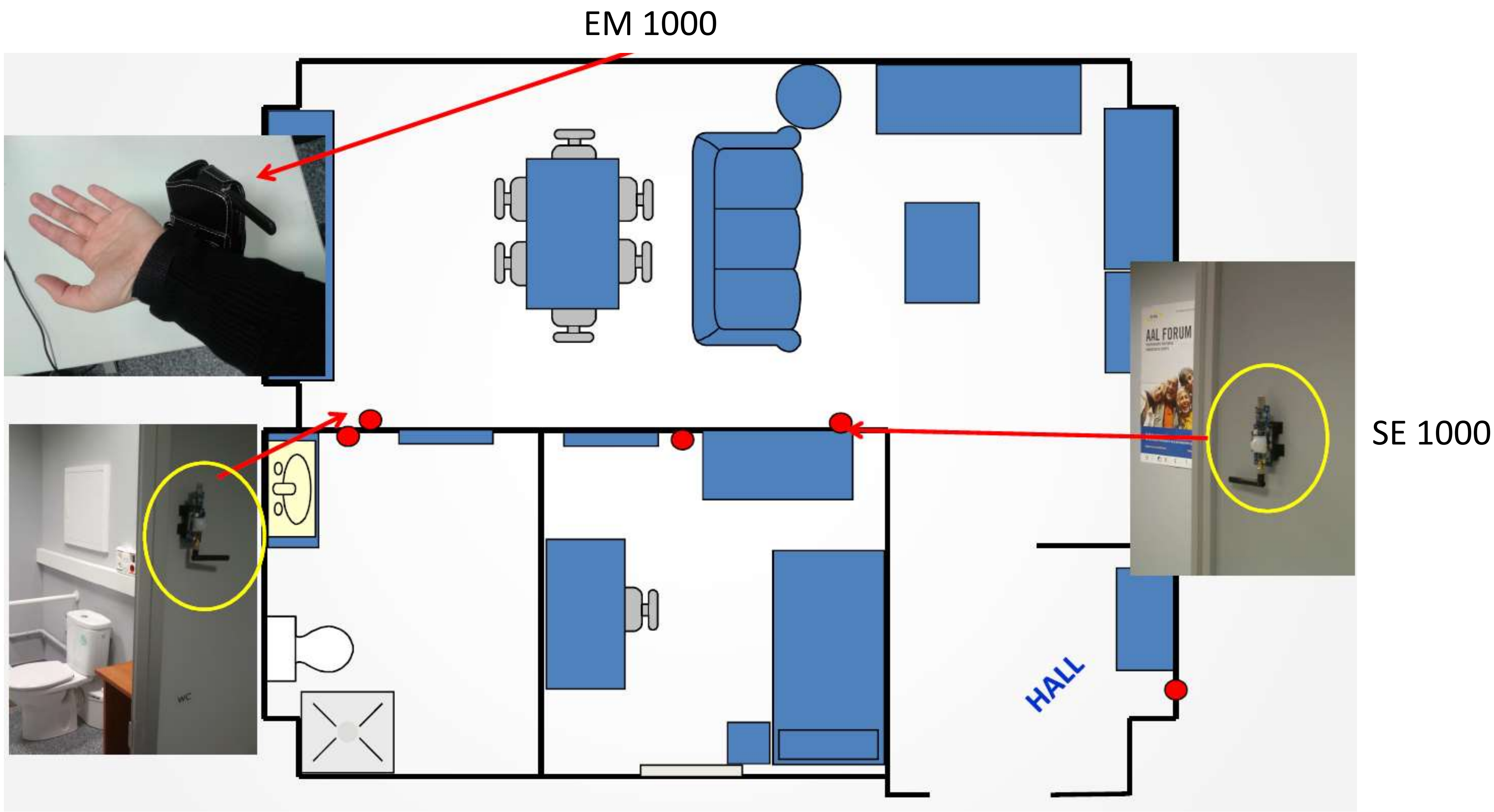
User Movement Forecasting



Graceful degradation with respect to the reservoir size

High accuracy on complex real-world data

Rubicon testbeds - AAL

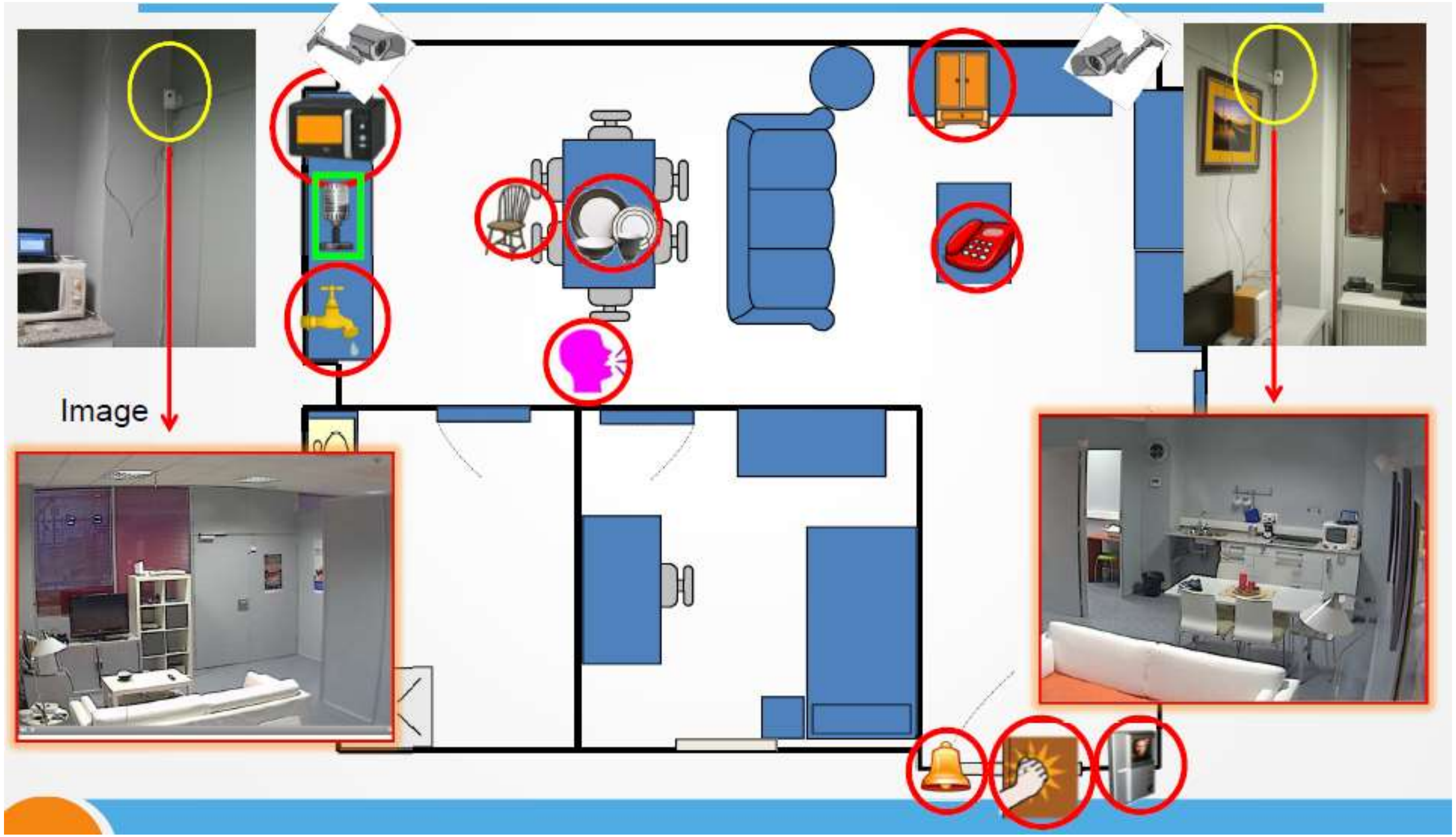


EM 1000

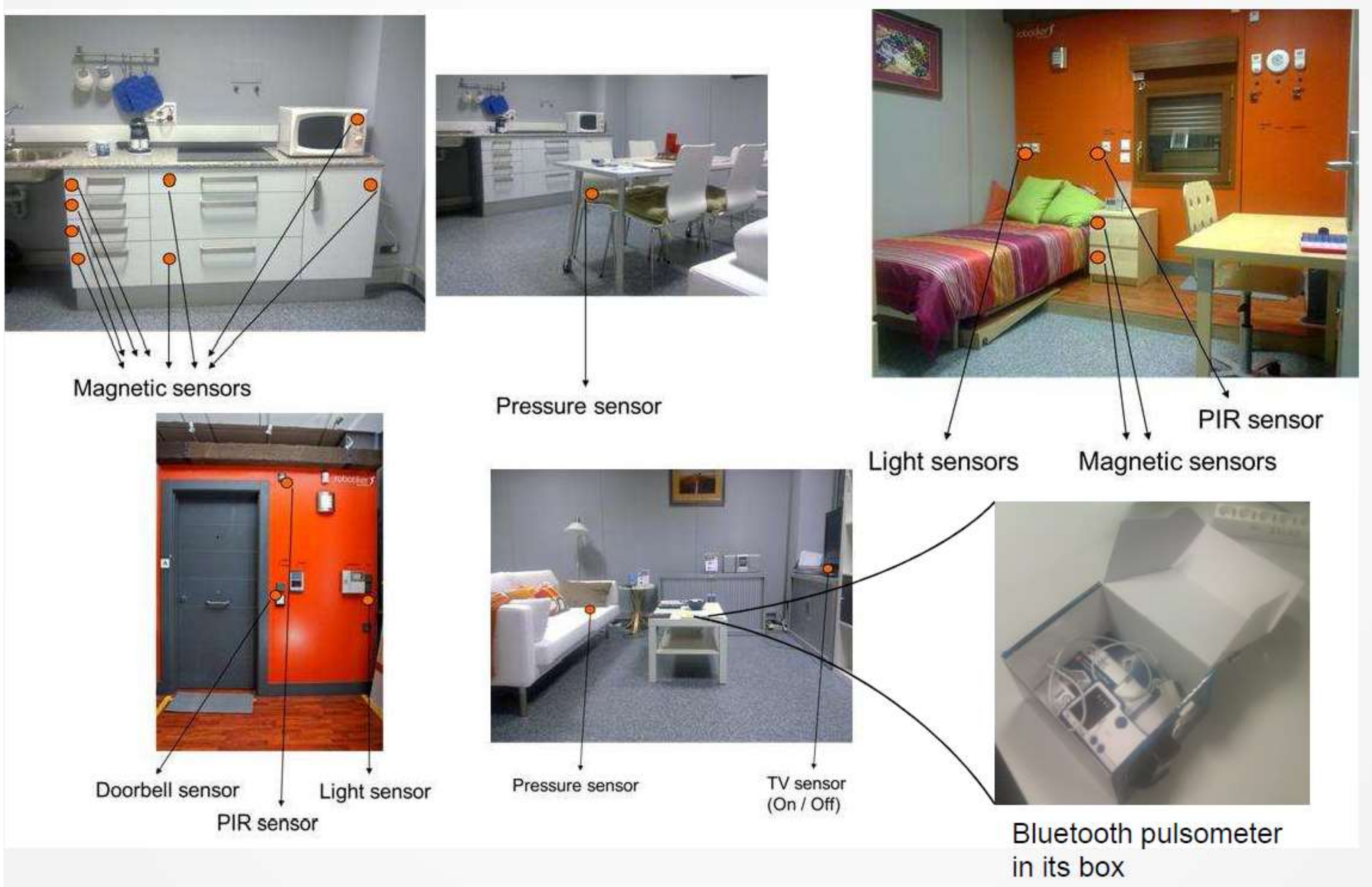
SE 1000

Rubicon testbeds - AAL

Sound recognition and cameras



Rubicon testbeds - AAL

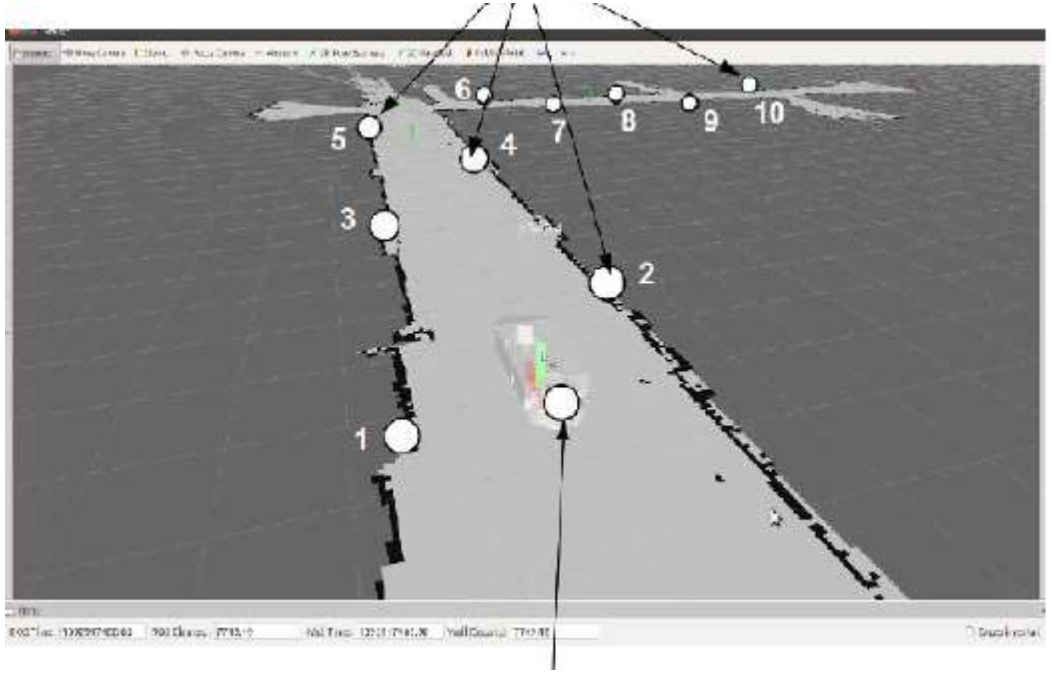


Rubicon testbeds - AAL

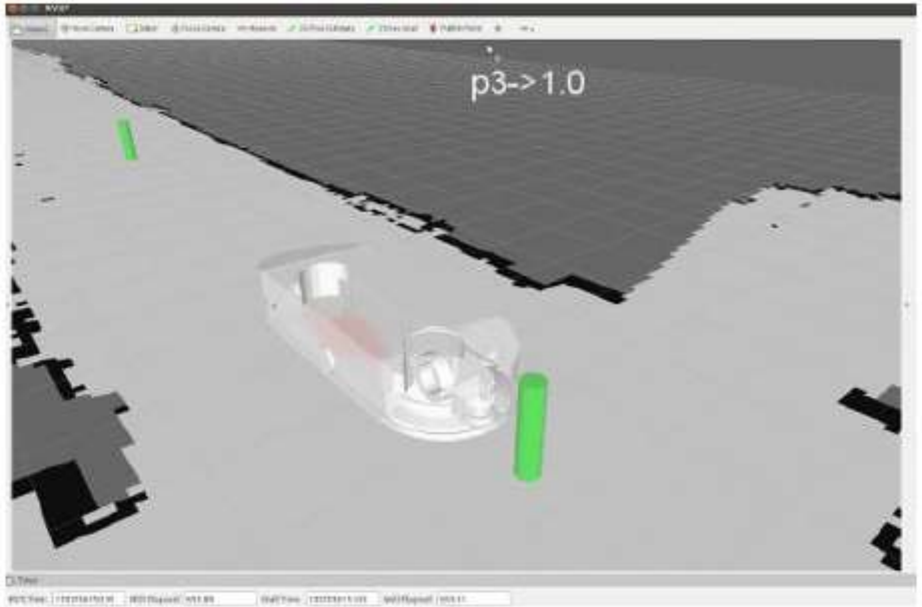
Rubicon use cases – robot navigation

Rubicon testbeds – robot navigation

RSSI anchor



Robot’s mote



doremi

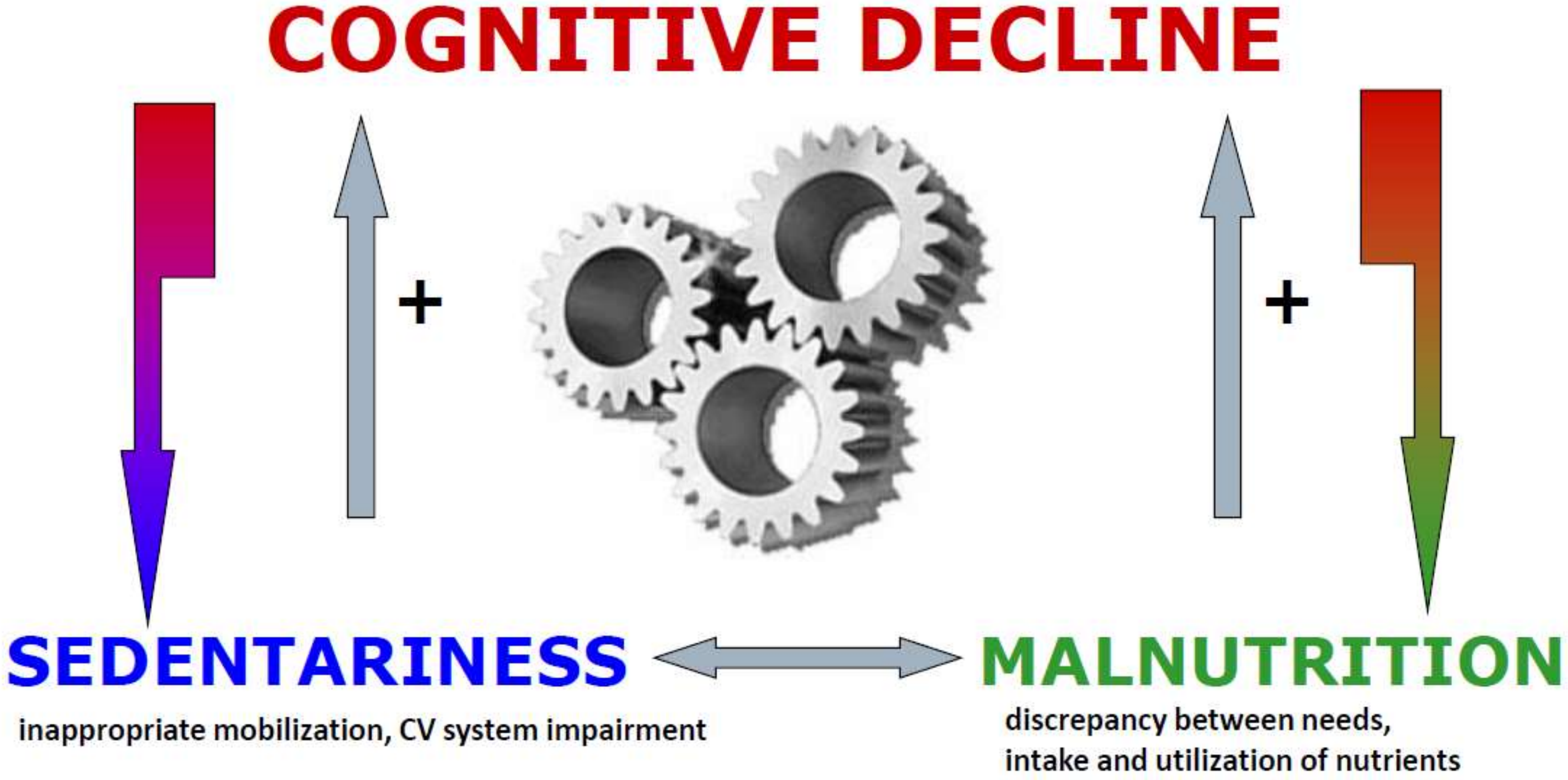
Decrease of cOgnitive decline, malnutRition and sedEntariness by elderly empowerment in lifestyle Management and social Inclusion

EU FP7 - http://www.doremi-fp7.eu/

General objectives

- Promote an active aging lifestyle
- Contrast:
  - Cognitive decline
  - Sedentariness
  - Malnutrition
- Use of ICT technologies:
  - Cognitive games
  - Physical and social activity monitoring
  - Diet monitoring

The Perverse Link among the 3 Impairments



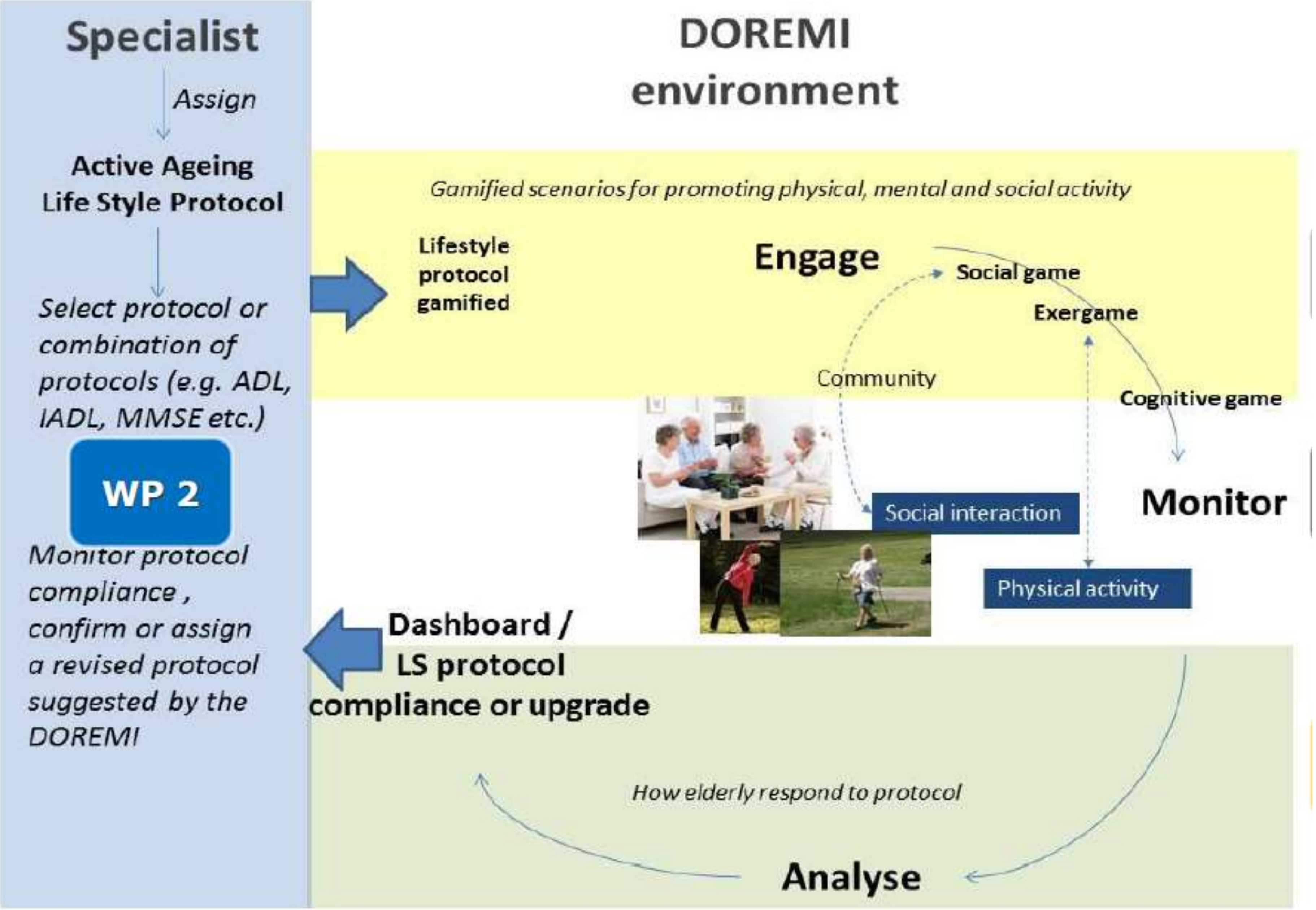
COGNITIVE DECLINE

SEDENTARINESS

inappropriate mobilization, CV system impairment

MALNUTRITION

discrepancy between needs,
intake and utilization of nutrients



Specialist

Assign

Active Ageing Life Style Protocol

Select protocol or combination of protocols (e.g. ADL, IADL, MMSE etc.)

WP 2

Monitor protocol compliance , confirm or assign a revised protocol suggested by the DOREMI

DOREMI environment

Gamified scenarios for promoting physical, mental and social activity

Lifestyle protocol gamified

Engage

Social game

Exergame

Community

Cognitive game

Social interaction

Monitor

Physical activity

Dashboard / LS protocol compliance or upgrade

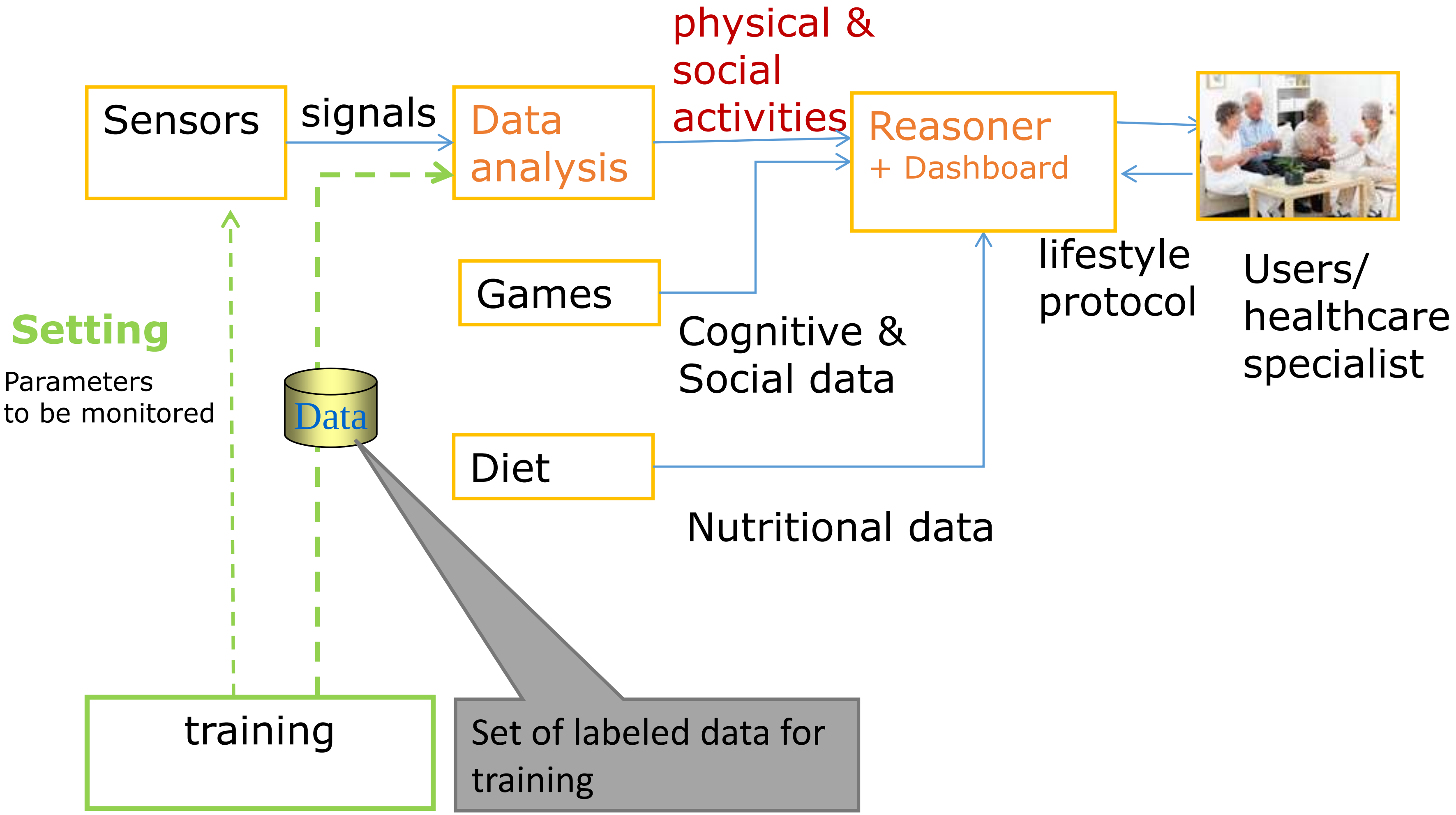
How elderly respond to protocol

Analyse

DOREMI - Activity recognition

- Specification of the user activities to be monitored by the HAR system (3 high level classes of HAR tasks)
  - Balance assessment
    - Aim: Estimation of user balance abilities in terms of membership to a stability class
    - Key inputs: DOREMI smart carpet sensors
  - Physical activity level
    - Aim: Quantify physical activity levels and associated energy expenditure
    - Key inputs: Accelerometers and heart rate data from the DOREMI bracelet
  - Social skills
    - Aim: People encountered estimation
    - Key inputs: Environmental sensors

DOREMI – activity recognition



Sensors

signals

Data analysis

physical & social activities

Reasoner + Dashboard

Users/ healthcare specialist

lifestyle protocol

Games

Cognitive & Social data

Diet

Nutritional data

Setting

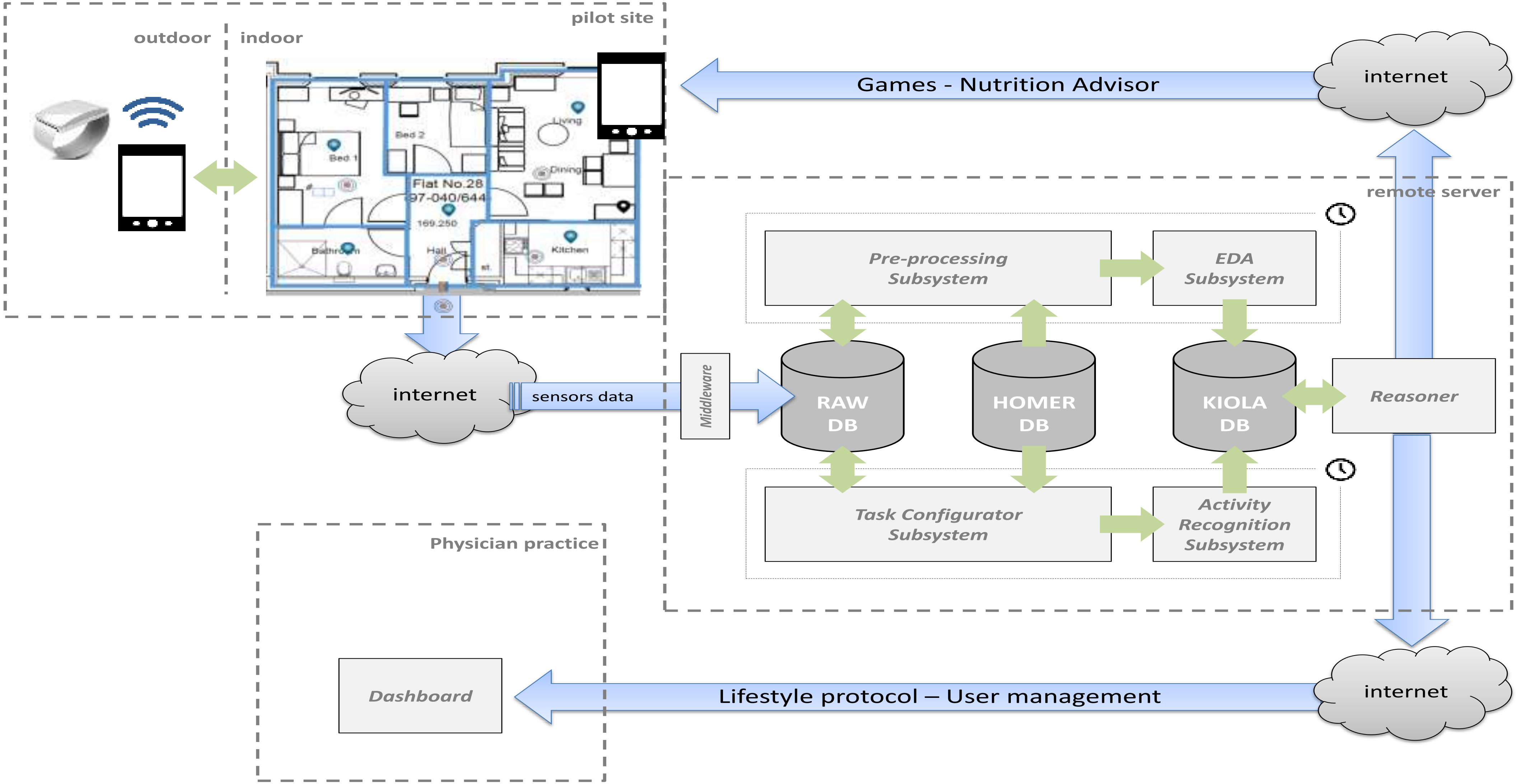
Parameters to be monitored

Data

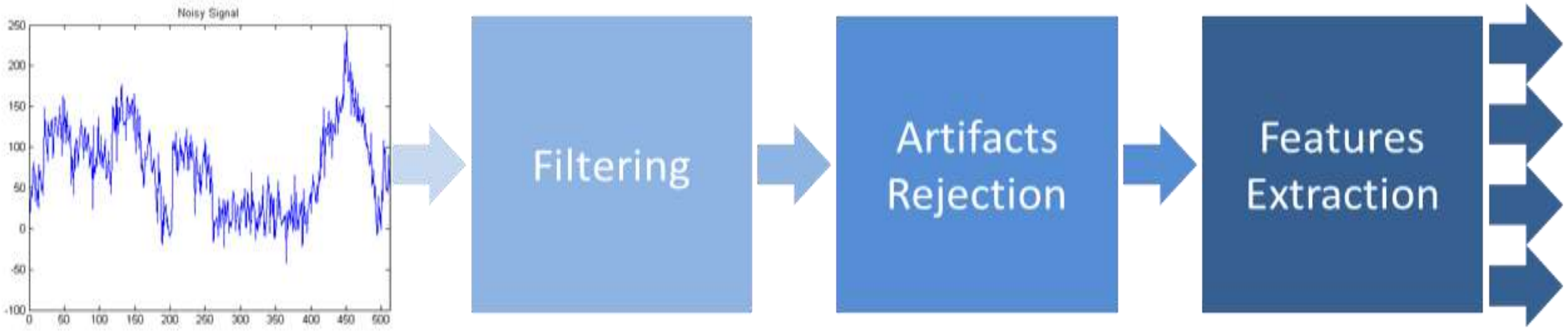
training

Set of labeled data for training

DOREMI - deployment



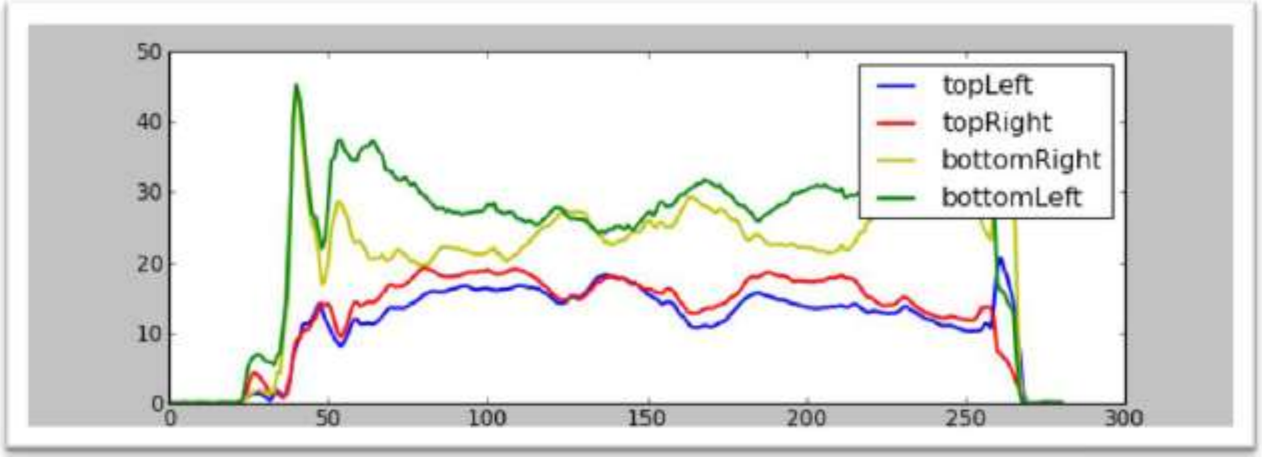
Preprocessing



- □ First set of features extracted from preliminary sensory data streams (sensors similar to those to be deployed in pilot sites)
  - □ Statistical features
  - □ Time series analysis features
  - □ Frequency domain features

DOREMI – the balance board

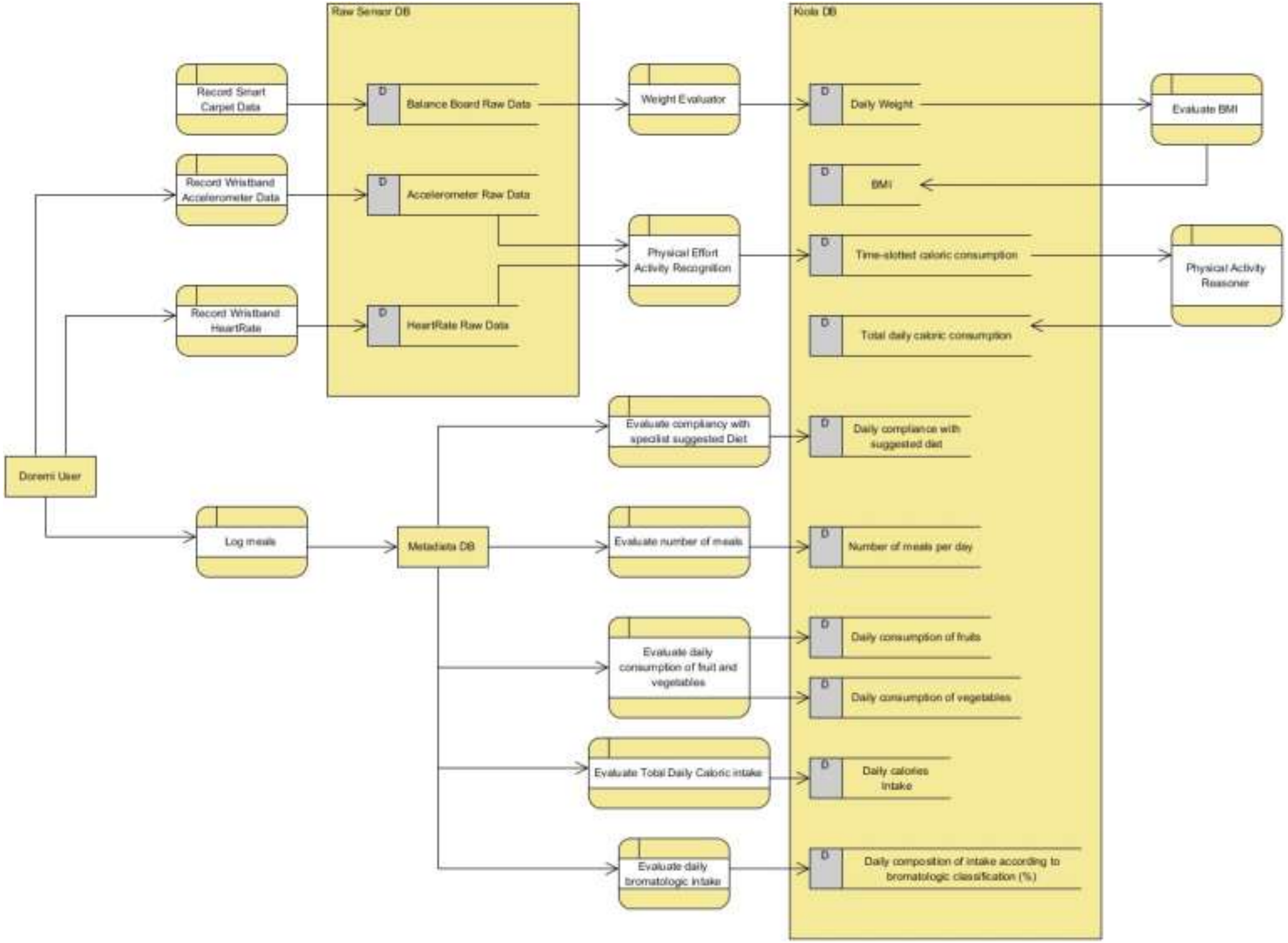
Berg Balance Scale (BBS) test #1: SITTING TO STANDING
INSTRUCTIONS: Please stand up. Try not to use your hand for support.
4 (x) able to stand without using hands and stabilize independently
3 ( ) able to stand independently using hands
2 ( ) able to stand using hands after several tries
1 ( ) needs minimal aid to stand or stabilize
0 ( ) needs moderate or maximal assist to stand



BBS assessment score: 55
Total estimated weight: 80 kg

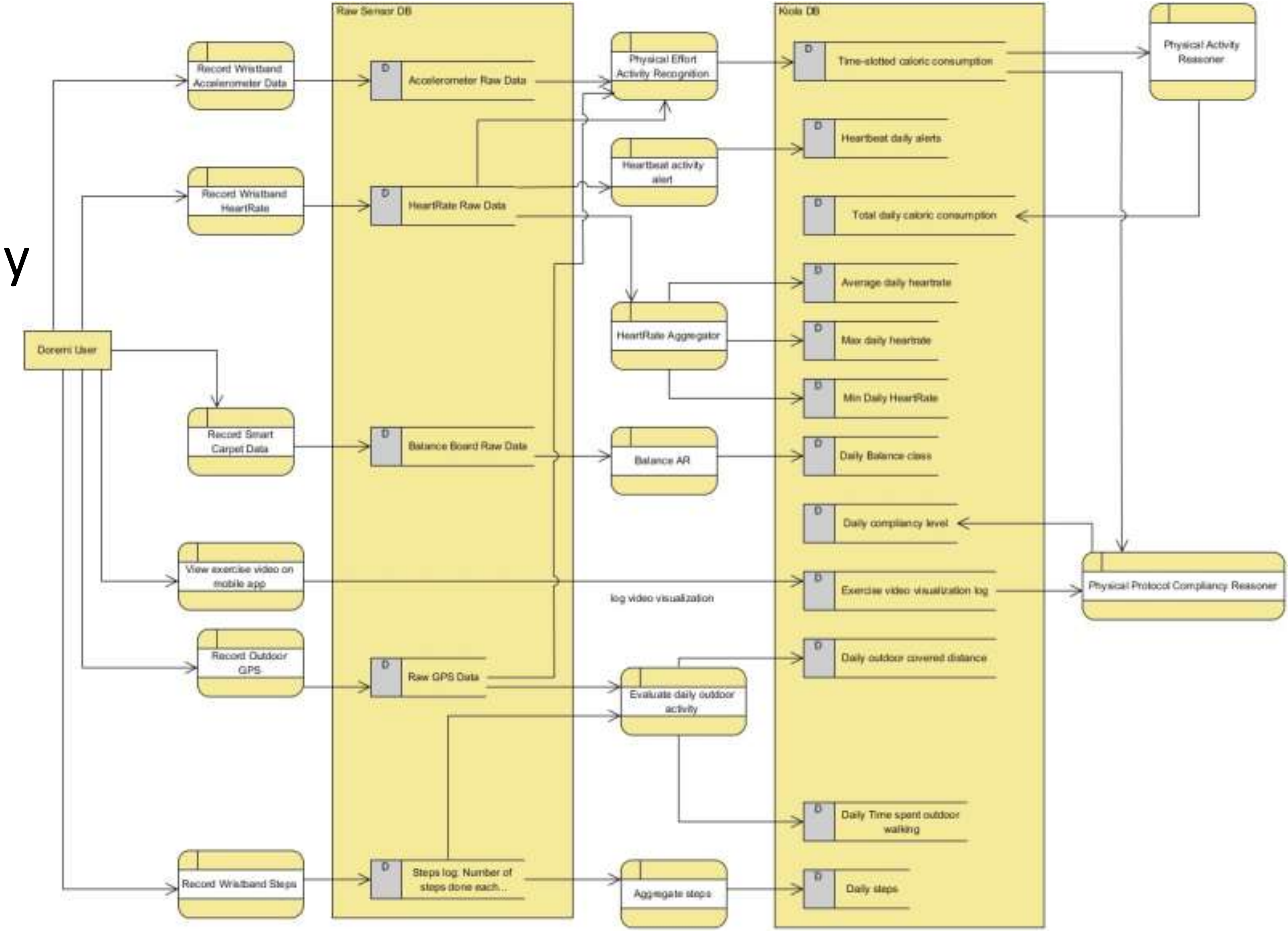
Dietary Data flow

- Compliancy with prescribed diet
- Number of meals
- Total caloric intake
- Daily consumption of fruit, vegetables
- Weight


Raw Sensor DB
Kiola DB
Record Smart Carpet Data
Record Wristband Accelerometer Data
Record Wristband HeartRate
Doremi User
Log meals
Metadata DB
D
Balance Board Raw Data
D
Accelerometer Raw Data
D
HeartRate Raw Data
Weight Evaluator
Physical Effort Activity Recognition
Evaluate compliancy with specilist suggested Diet
Evaluate number of meals
Evaluate daily consumption of fruit and vegetables
Evaluate Total Daily Caloric intake
Evaluate daily bromatologic intake
D
Daily Weight
D
BMI
D
Time-slotted caloric consumption
D
Total daily caloric consumption
D
Daily compliance with suggested diet
D
Number of meals per day
D
Daily consumption of fruits
D
Daily consumption of vegetables
D
Daily calories intake
D
Daily composition of intake according to bromatologic classification (%)
Evaluate BMI
Physical Activity Reasoner

Sedentariness data flow

- Outdoor distance
- Number of steps per day
- Heart rate
- Balance
- Physical activity recognition


Raw Sensor DB
Kiola DB
Doremi User
Record Wristband Accelerometer Data
Record Wristband HeartRate
Record Smart Carpet Data
View exercise video on mobile app
Record Outdoor GPS
Record Wristband Steps
Accelerometer Raw Data
HeartRate Raw Data
Balance Board Raw Data
Raw GPS Data
Steps log: Number of steps done each...
Physical Effort Activity Recognition
Heartbeat activity alert
HeartRate Aggregator
Balance AR
log video visualization
Evaluate daily outdoor activity
Aggregate steps
Time-slotted caloric consumption
Heartbeat daily alerts
Total daily caloric consumption
Average daily heartrate
Max daily heartrate
Min Daily HeartRate
Daily Balance class
Daily compliancy level
Exercise video visualization log
Daily outdoor covered distance
Daily Time spent outdoor walking
Daily steps
Physical Activity Reasoner
Physical Protocol Compliancy Reasoner

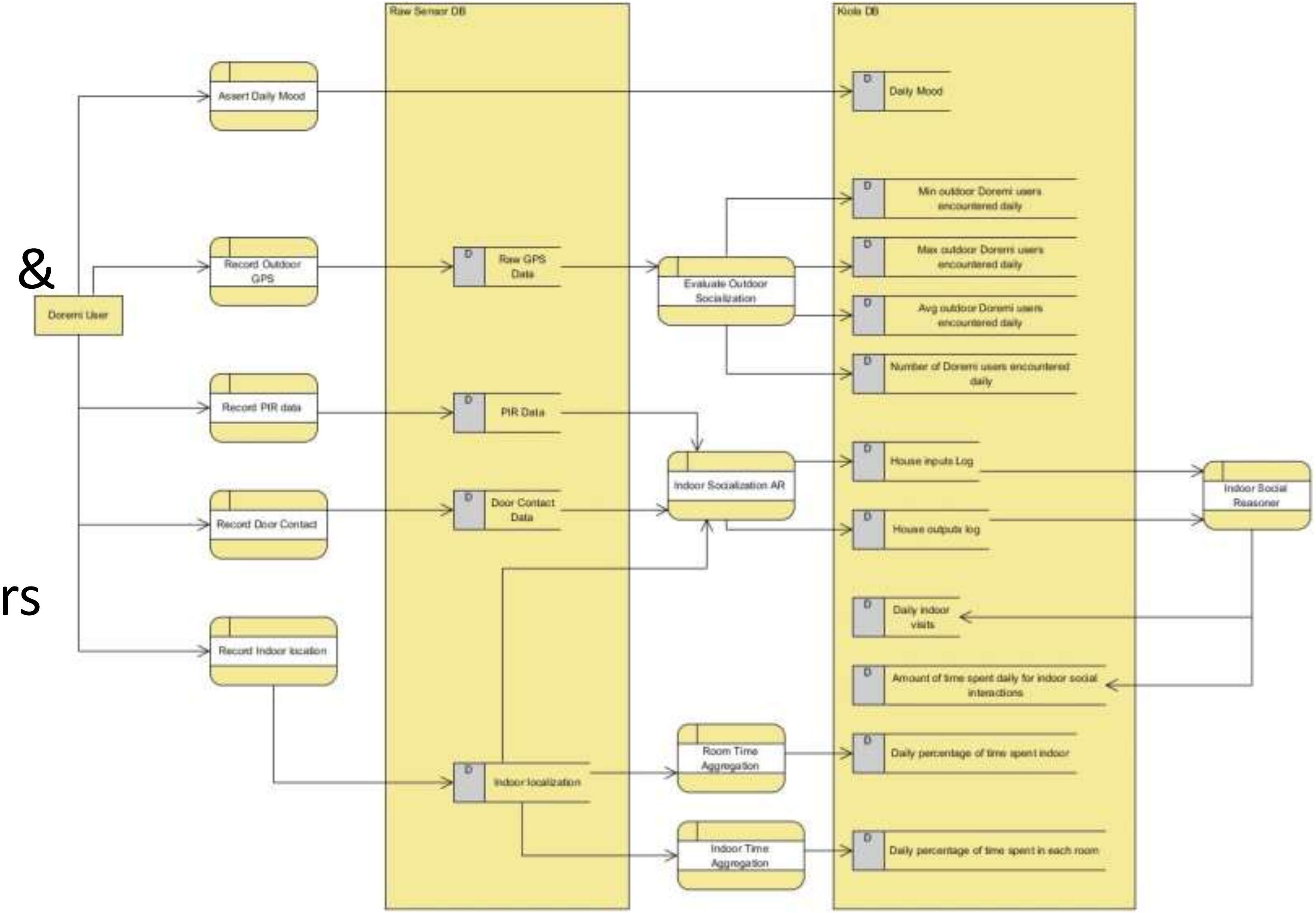
Social & cognitive Data Flow

Social:

- Number of people met & contact duration
- Time spent indoor

Cognitive:

- number of right anwsers
- Reaction time
- …



Hardware platforms

Mica Motes

Mica Z

a)

Iris

Cricket

AdvanticSys Mote CM 5000

Sensor network hardware

The Mica2/MicaZ platform:

- Low power CPU
  - ATMEL 128L (8 bit, 8Mhz)
- Program memory: 128 KB Flash memory
- Data memory: 4 KB RAM – 512 KB Flash memory

MICA2 Board

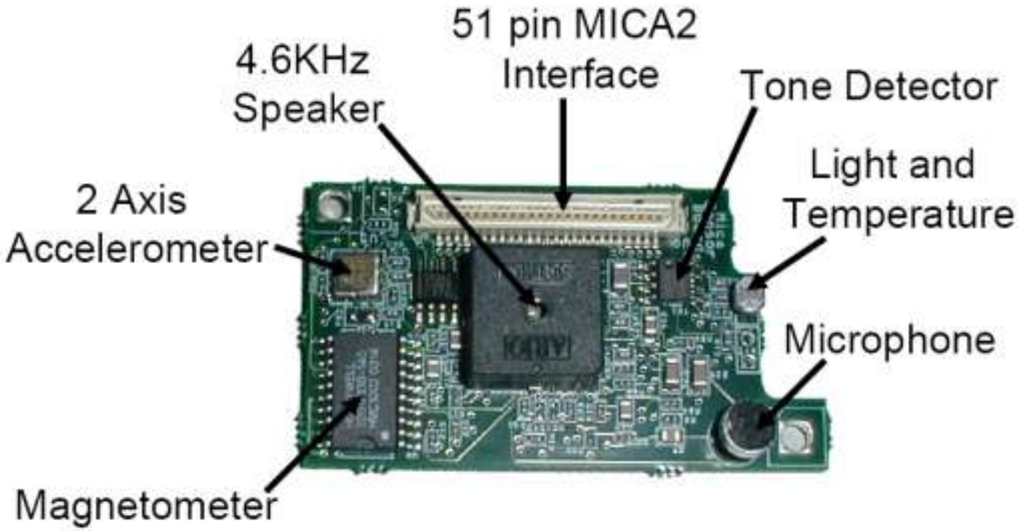
MICA2dot Board

Mica Motes: transducer board

- Example: MTS 300 CA
  - Light
  - Temperature
  - Microphone
  - Sounder
  - Accelerometer 2 axis
  - Magnetometer 2 axis

Other boards include:

- GPS
- Humidity
- Pressure
- Additional analog and digital inputs



Other sensor boards for AdvanticSyS

Passive infrared

CO/CO2, dust

Pressure & vibration

An introduction to Arduino

Content

- **Introduction on Arduino world;**
  - Idea of Arduino project;
  - “Arduino” employment;
- **Arduino: the device;**
  - Models of devices;
  - Models enable for your projects;
  - Technical characteristics;
  - Device characteristics;
  - Sensors;
- **Arduino: development environment;**
  - How to prepare the environment;
  - IDE;
  - Sketch and its structure;
  - Language and libraries;
- **Arduino: Support;**
  - Libraries;
  - Forum and Support;
  - Interesting projects;
- **Examples;**
- **Try it;**

The Idea of Arduino

Arduino is an **open-source electronics prototyping**

platform based on flexible,

easy-to-use hardware and software.

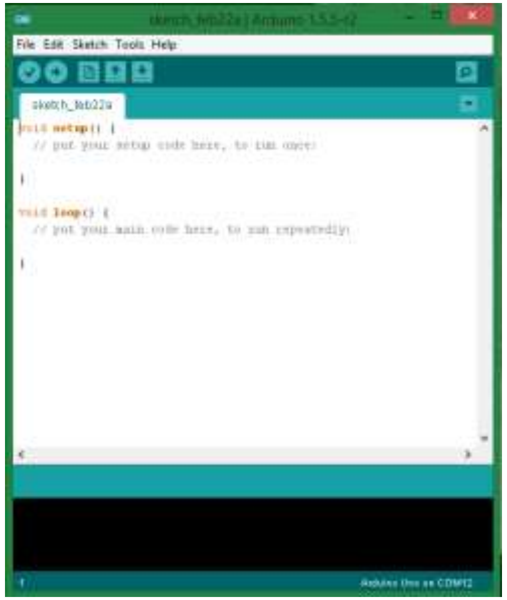
It's intended for artists, designers, hobbyists and anyone interested in **creating interactive objects or environments.**

“Arduino”

“Arduino” is:

- Device

- IDE



```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

Arduino Uno on COM12

- Forum

Using Arduino

Installation & Troubleshooting
For problems with Arduino itself, NOT your project
Last post: [illegible] on Today at [illegible]
34946 Posts 7005 Topics

Project Guidance
Advice on general approaches or feasibility
Last post: [illegible] on Today at [illegible]
152931 Posts 2171 Topics

Programming Questions
[illegible]
320386 26250

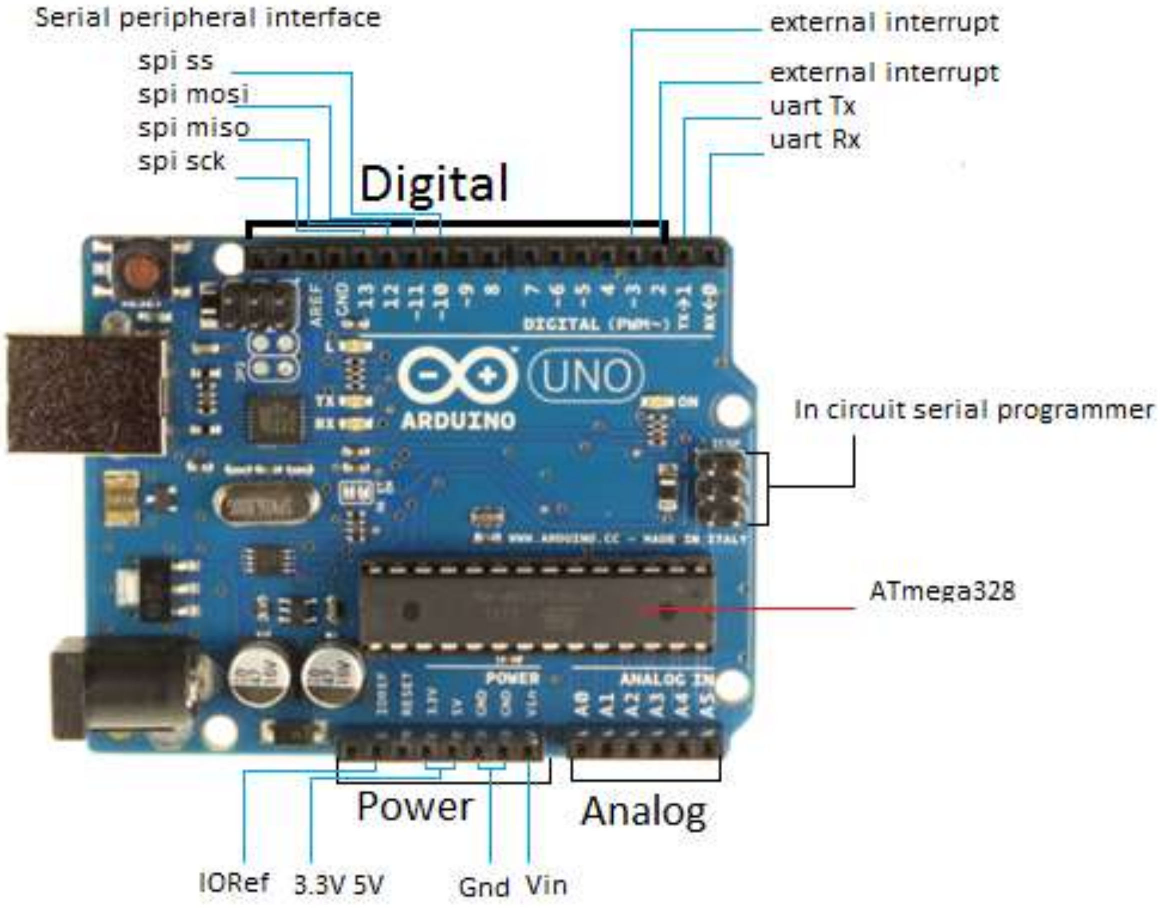
Hardware: some models

UNO

YÚN

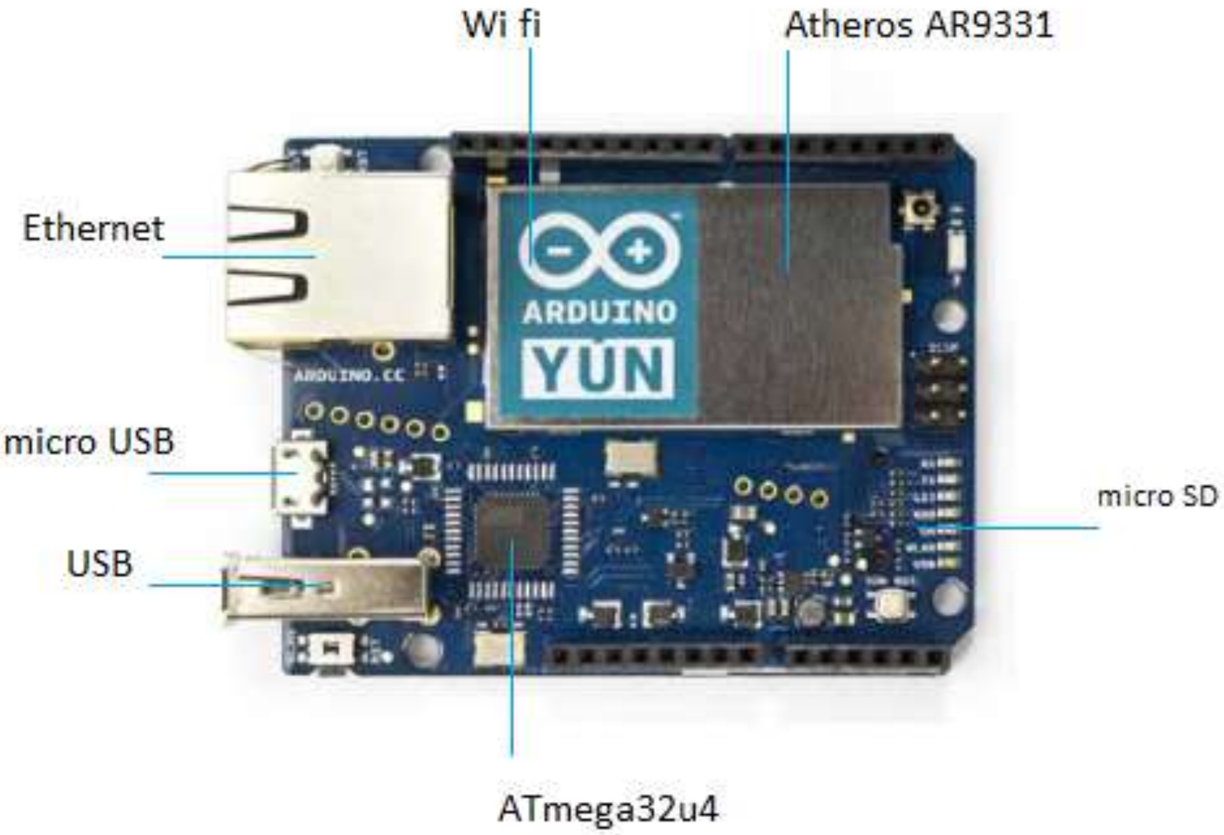
Arduino UNO

- **AVR Arduino microcontroller**
  - **Atmega328**
    - SRAM 2KB
    - EEPROM da 1KB
    - Flash memory 32 KB



Arduino YÚN

- **AVR Arduino microcontroller**
  - **Atmega32u4**
    - Flash memory 32 Kb
    - SRAM 2.5KB
    - EEPROM 1KB
- **Linux microprocessor**
  - **Atheros AR9331**
    - RAM 64 MB DDR2
    - 16MB Flash memory



Sensors, Actuators, and Shields

- Sensors
  - Accelerometer module
  - Tilt module
  - Button module
  - Linear potentiometer
  - Rotatory potentiometer
  - Joystick module
  - Hall sensor module
  - LDR sensor module
  - Temperature sensor module
  - Touch sensor module
  - Humidity sensor
  - GPS module
  - Piezo

- Actuators
  - Led (red, blue, green, yellow)
  - Power Led module
  - Servo motors
  - Stepper motors
  - Paper panel

- For high power
  - Mosfet module
  - Relay module

- Shields
  - Bluetooth
  - GSM
  - Zigbee

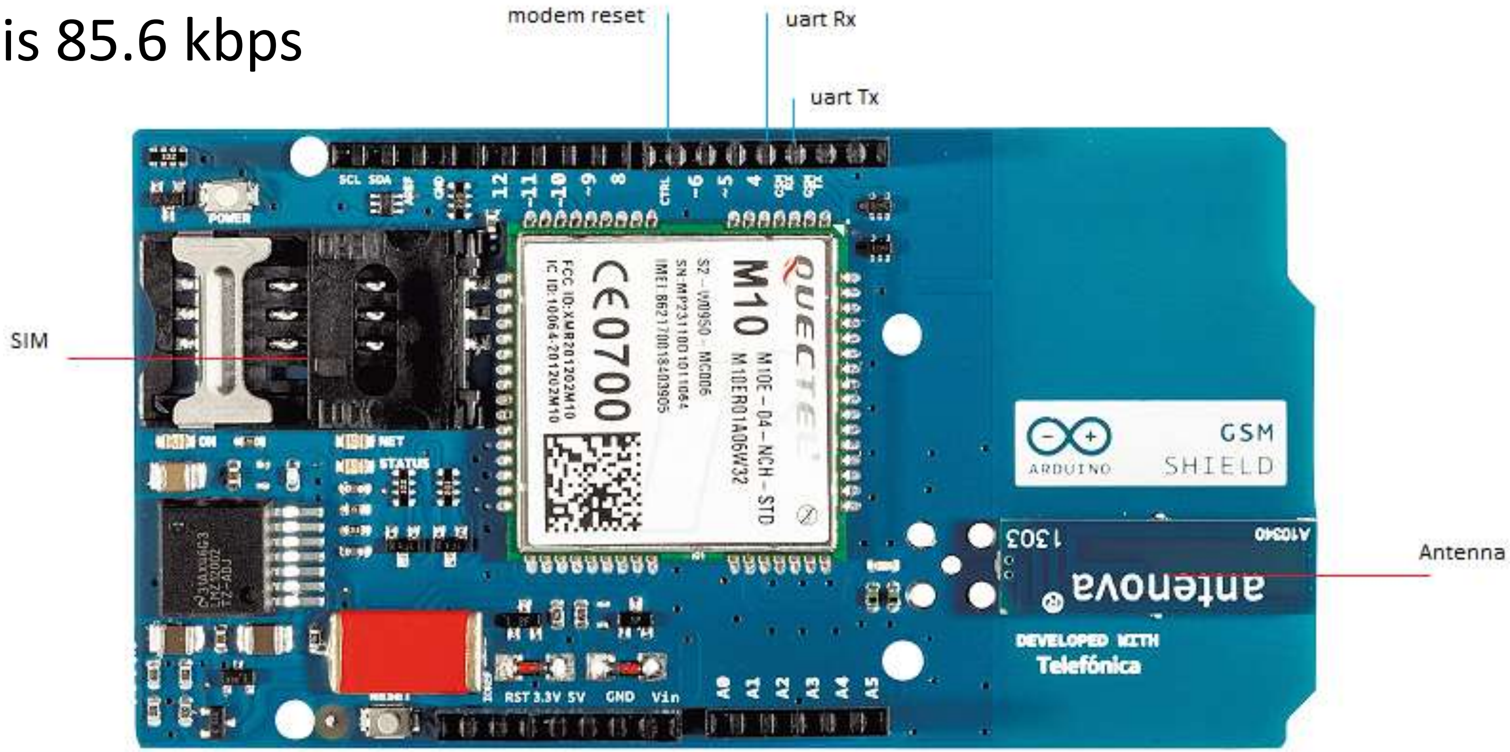
Bluetooth and Xbee module

- Bluetooth® version 2.1 module
- It supports the EDR (Enhanced Data Rate)
- Delivers up to a 3 Mbps data rate for distances up to 20 meters

- Xbee module series 1
- Standard 802.15.4
- Set as coordinator, router, end node
- 250kbps Max data rate
- 100m range

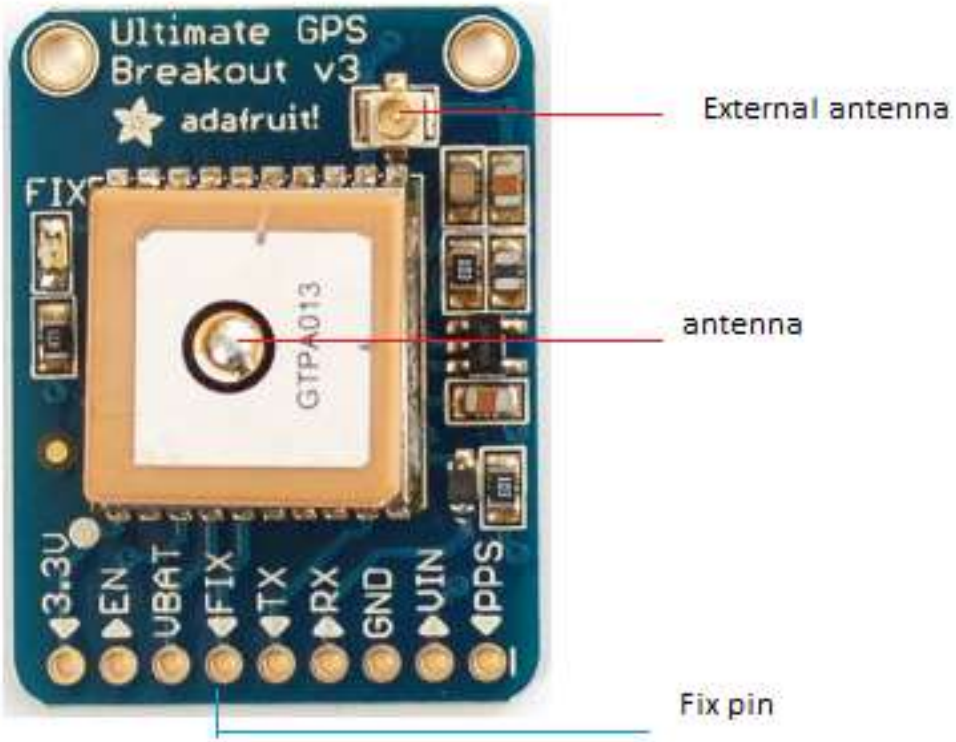
GSM shield

- Quad-band GSM/GPRS modem
- Supports TCP/UDP and HTTP
- Speed maximum is 85.6 kbps



GPS module

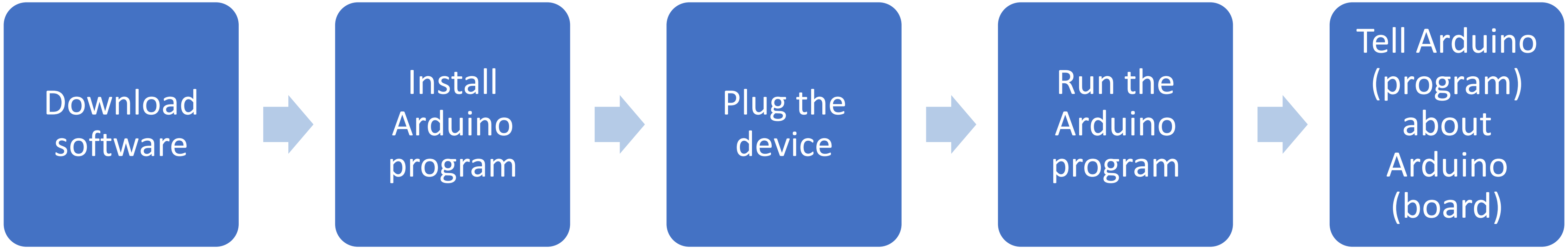
- P
- Low power requirements
- Ultra-low dropout 3.3V regulator so you can power it with 3.3-5VDC in, 5V level safe inputs
- Position accuracy of 1.8 meters
- Velocity accuracy of .1 meters per second



Software: how to prepare the environment

The **open-source** Arduino **environment** makes it **easy to write code** and **upload it to the I/O board**. It runs on **Windows, Mac OS X, and Linux**. The environment is written in Java and based on Processing, avr-gcc, and other open source software.

Arduino IDE can be downloaded at www.arduino.cc



Download software

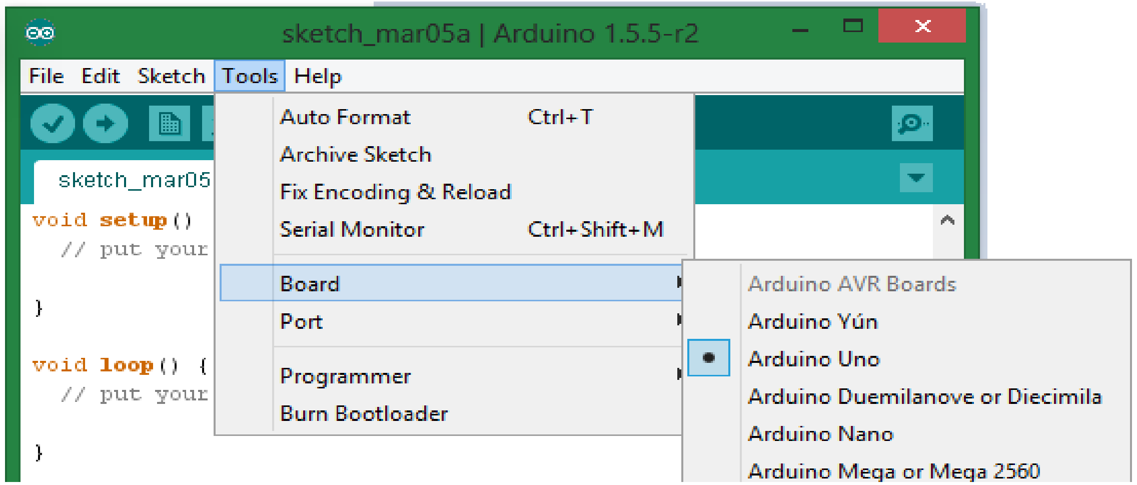
Install Arduino program

Plug the device

Run the Arduino program

Tell Arduino (program) about Arduino (board)

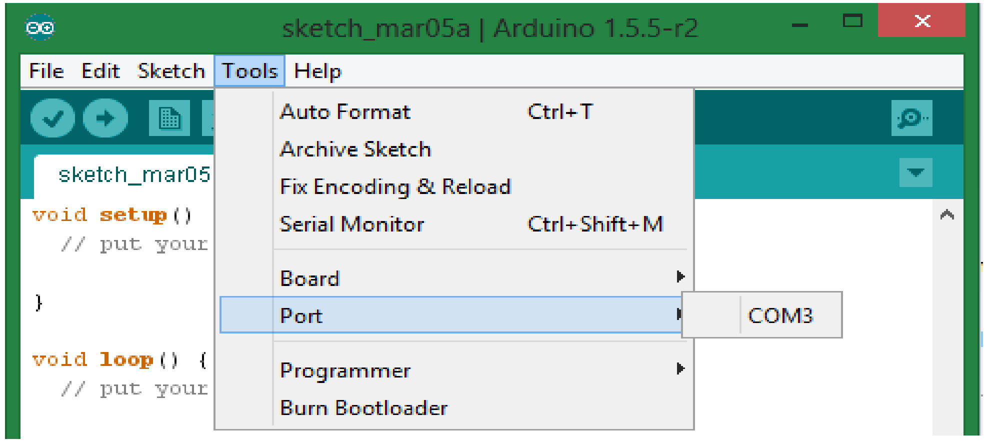
Selection Location and Type



● Arduino Uno

```
void setup()
  // put your

}

void loop() {
  // put your

}
```

Select your arduino



COM3

```
void setup()
  // put your

}

void loop() {
  // put your
```

Select the location of device

Terminology

- “*sketch*” – a program you write to run on an Arduino board
- “*pin*” – an input or output connected to something.
  - e.g. output to an LED, input from a knob.
- “*digital*” – value is either HIGH or LOW.
  - (aka on/off, one/zero) e.g. switch state
- “*analog*” – value ranges, usually from 0-1023.
  - e.g. LED brightness, motor speed, etc.

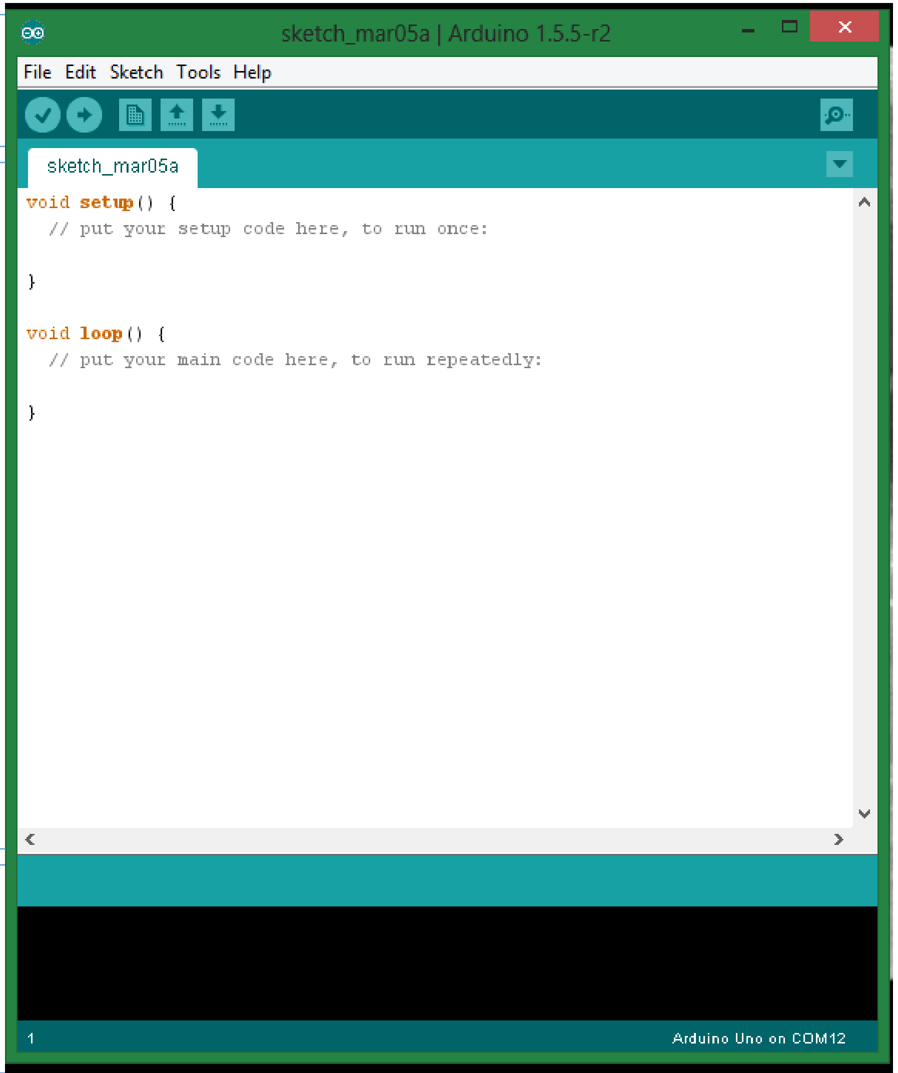
IDE

- Verify
- Upload
- New
- Open
- Save
- Serial monitor

Toolbar buttons

Sketch editor

Console display

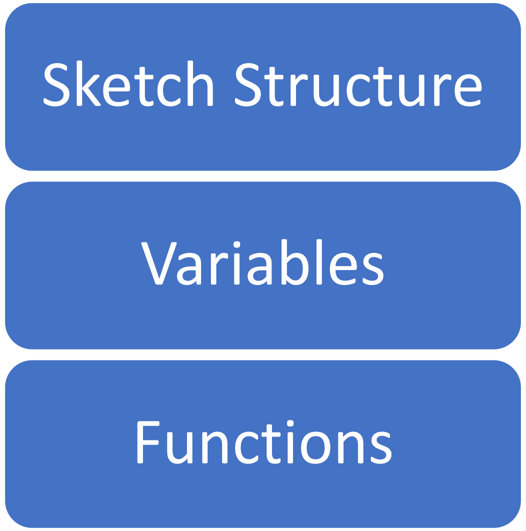


```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

Language

The Arduino environment is based on Atmel Atmega microcontrollers. The AVR language is a "C" environment for programming Atmel chips.

The programs can be divided in three main parts:

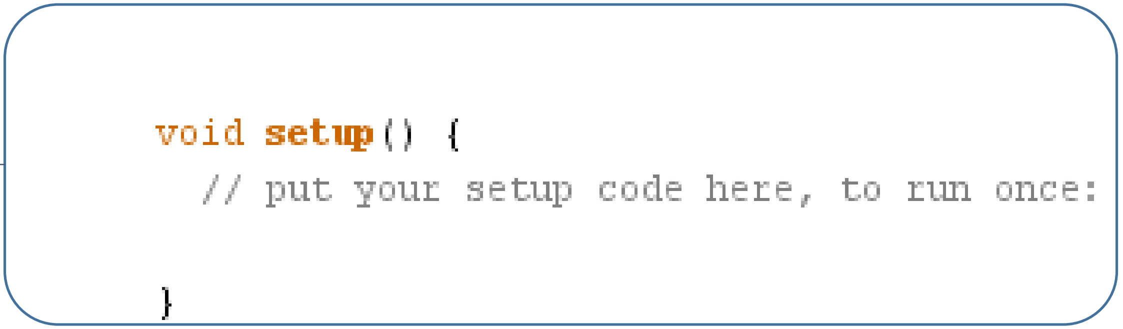


Sketch Structure

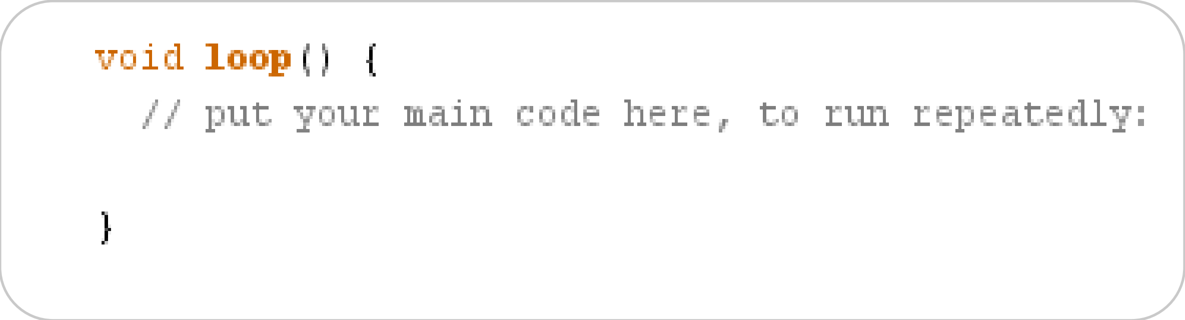
Variables

Functions

Sketch and its structure



```
void setup() {
  // put your setup code here, to run once:

}
```



```
void loop() {
  // put your main code here, to run repeatedly:

}
```

Called when a sketch starts.
The setup function will only run once.

Does precisely what its name suggests,
and loops consecutively.

Other structure functions

- Control Structures: if then else, for, switch, while, continue, return, goto …;
- Further Syntax: ;, {}, //, /**/, #include, #define;
- Arithmetic Operators: +, -, =, /, *, %;
- Comparison Operators: ==, !=, <, >, <=, >=;
- Boolean Operators: &&, ||, !;
- Pointer Access Operators: *, &;
- Bitwise Operators: &, |, ^, >>, <<, ~;
- Compound Operators: ++, --, ==, +=, -=, *=, /=, &=, |=;

Variables

- Constants: level of energy (HIGH; LOW); mode of pin(INPUT; OUTPUT; INPUT_PULLUP); led13(LED_BUILTIN);…;
- Types: word; String;…;
- Conversions: word();…;
- Variable scope and qualifiers: Volatile;…;
- Usefulness: sizeof();

Functions

Functions are distinguished according to the pin:

- Digitals: pinMode(); digitalRead(); digitalWrite();
- Analogs: analogReference(); analogRead(); analogWrite();
- Advanced I/O: tone(); noTone(); shiftOut(); shiftIn(); pulseIn();
- Time: millis(); micros(); delay(); delayMicroseconds();
- Math: min(); max(); abs(); ...;
- Trigonometry: sin(); cos(); tan();
- Random Numbers: randomSeed(); random();
- Bits and Bytes: lowByte(); highByte(); bitRead(); bitWrite(); bitSet(); bitClear(); bit();
- External Interrupts: attachInterrupt() detachInterrupt()
- Interrupts: interrupts(); noInterrupts();
- Communication: Serial; Stream;

Libraries

All Libraries for all Arduino shields and components are on:

http://www.arduino.cc/en/Reference/Libraries

Forum & Support

Support for arduino programmer: http://forum.arduino.cc

Tutorial of Arduino Owner:

Arduino Tutorial

Starter projects with Arduino:

Starter Projects

Tutorial for AdaFruit component:

- GSM and GPS
- Adafruit products

Interesting projects

- Bare Conductive
- Smart citizen kit
- Little Robot Friends
- Little Bits
- Primo
- Earth Make
- Annikken Andee

ARDUINO AT HEART

Let’s try it

- Blink Led
- Potentiometer rotary + blink led
- Humid + Term with yun
- Volatile Button
- GPS paring