Wearable Computing Systems based on Body Sensor Networks: State-of-the-art and Future Research Challenges

Giancarlo Fortino

Full Professor of Computer Engineering @
Department of Informatics, Modeling, Electronics and Systems (DIMES)
University of Calabria, Rende (CS), ITALY

High-end Foreign Expert in China
Distinguished Professor at Huazhong Agricultural University (China)
Guest/Adjunct Professor of Computer Engineering at Wuhan University of Technology (China)
Adjunct Senior Research Fellow at Italian National Research Center (CNR – ICAR), Italy
High-end Expert at Huazhong University of Science and Technology (China)
CAS PIFI Visiting Scientist at SIAT – Shenzhen (China)
Director of the SPEME Lab and Co-Director of Joint China-Italy IoT Labs (Wuhan & Shanghai)
Co-founder and CEO of SenSysCal S.r.l. (www.sensyscal.com)

g.fortino@unical.it
http://wwwinfo.deis.unical.it/fortino/index.htm
http://labs.dimes.unical.it/speme/

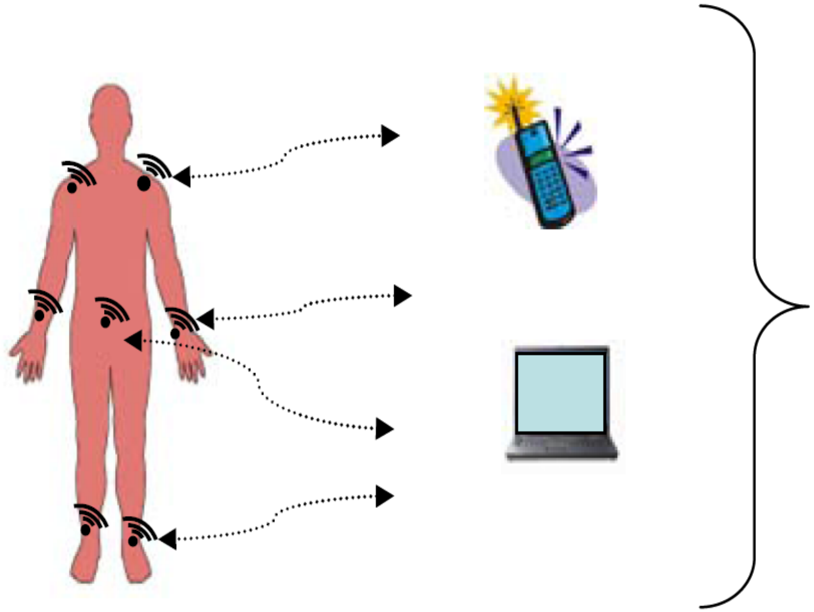
Outline

1. Wireless Body Sensor Networks (WBSN)
2. Embedded Frameworks for BSNs: State-of-the-Art
3. The SPINE Framework
   - Architecture and Evaluation
   - Applications
   - Demo
4. SPINE Variants/Extensions:
   - C-SPINE, A-SPINE, SPINE2, SPINE-*
5. Conclusions
6. Research Challenges and Future Work

WBSNs: Background

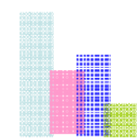
- A **Wireless Body Sensor Network** is a Wireless Sensor Network applied to the human body.
- Typically, multiple sensors are worn on the body, communicating wirelessly to a local coordinator station (and possibly remotely to a central server)
- Applications:
  - **health-care**,
  - **fitness and wellness**,
  - **emotion and stress recognition**,
  - **interactive gaming**.
  - **social interaction**.

WBSNs support a range of life-aiding applications



Activity Monitoring

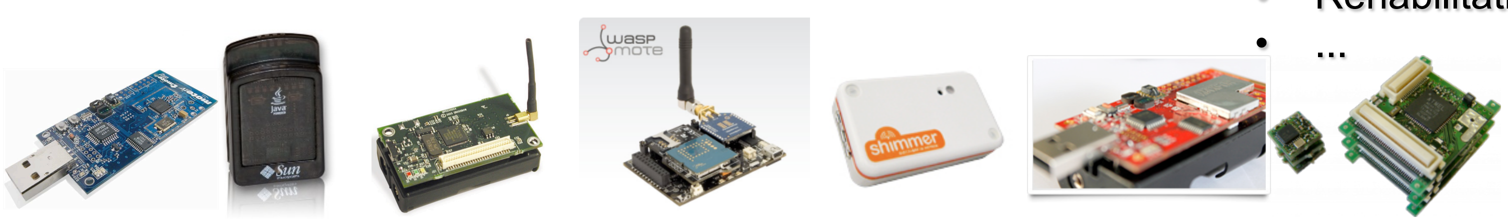
Fall Detection



Statistics about movements

- Gesture recognition
- Emotional stress analysis
- Handshake detection
- Rehabilitation
- ...

Body Sensor Network ⟷ **Gateways** ⟷ **Services**



WBSNs: research motivation

- Programming BSN systems is a complex task
  - **Constrained hardware resources**;
  - (quasi) real time requirements;
  - **Low-level software abstractions** (languages, native libraries, OSs);
  - Signal-processing and decision support;
  - Security and privacy.

WBSN Programming: Related Work

- Three main approaches:
  - Application-specific development
    - Greatest majority of the WBSN research prototypes so far
  - General-purpose middlewares for WSNs
    - **Titan** (customized for WBSNs), **MAPS** (exploited for BSN applications, http://maps.deis.unical.it)
  - Domain-specific frameworks
    - **CodeBlue** (very limited signal processing integrated support)
    - **RehabSPOT** (limited customization, only works on SunSPOT)

Objective

- **Application-specific development** is time consuming, error-prone, and produces code scarcely reusable
- **General purpose middlewares** for WSNs do not address specific BSN system needs

*To fully address such limitations, our research focused on defining effective **domain-specific** design methodologies and embedded frameworks for programming signal-processing intensive WBSN applications.*

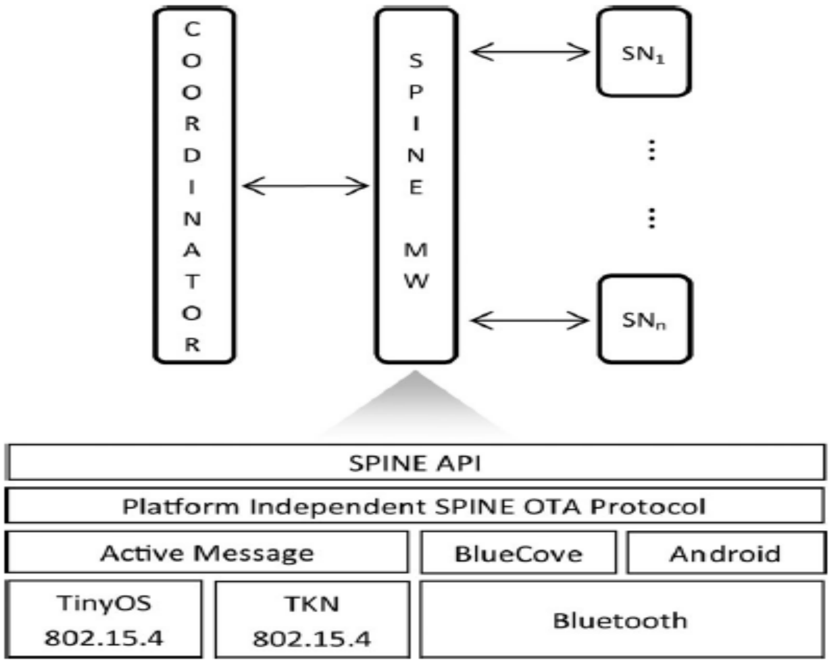
WBSN Programming: Requirements

TASK	DESCRIPTION
SENSOR SAMPLING	The sensor sampling process represents the first step for developing a BSN application. Selecting the appropriate sampling time to satisfy the application requirements is important, as it determines the amount of raw data generated and processed (and to a certain degree, energy consumed). The proper execution of the application may depend on this parameter; often, a minimum sampling time is required to allow a sensor to accurately capture a particular phenomena.
IN-NODE DATA PROCESSING	Classifier algorithms very rarely use raw data. Instead, attributes (or features) are extracted on sample data windows and used to detect events and classify activities. Extracting features directly on the wireless nodes allows for reduction of radio usage, as resulting summary data are sent instead of raw data values.
SENSOR CONFIGURATION AT RUN-TIME	Support for runtime configuration (enabling, disabling, setting the sampling rate) on the available sensors of a node is often very useful. Application requirements can change over time; for instance, under certain circumstances, a sensor may be sampled at a lower rate, or its data not needed at all. Therefore, supporting runtime sensor configuration allows dynamic application behavior.
NODE SYNCHRONIZATION	Many BSN-oriented signal-processing algorithms require sensors on multiple nodes to be sampled in unified time intervals, to ensure consensus of time in observing underlying events. Nodes are often kept synchronized to in turn allow synchronized sampling of sensors and joint processing data at the coordinator.
DUTY-CYCLING	Duty cycling is a mechanism for controlling radio power, to reduce power consumption of a sensor node, thus increasing its battery lifetime. Radio duty cycling must be tuned very carefully in order to minimize energy use, but allow sufficient transmission of data.
APPLICATION LEVEL COMMUNICATION PROTOCOL	A specific application level communication protocol is needed to support the interaction among sensor nodes (if needed) and between sensor nodes and the coordinator. The communication involves sensor node discovery and service advertisement, requests for sensing and signal processing, raw and preprocessed sensor data transmission, and event delivery.
HIGH-LEVEL PROCESSING	Often, the end-user BSN applications do much more than plotting sensor data into graphs. They require the interpretation of asynchronous events and periodic data coming from sensors in high-level knowledge. This implies decision support (classification) algorithms that extract meaningful information from such events and data.

REQUIREMENT	HIGH-LEVEL TECHNIQUES
Programming Effectiveness	Programming Abstractions, Software Engineering Methods, Debugging and Testing Tools
System Efficiency	Resource Management Optimization
System Interoperability	Application-Level Communication Protocol and Adapter for Heterogeneous Sensor Inclusion
System Usability	GUI-based flexible management of the BSN system, PC and smartphone-based Coordinator
Privacy Support	Data encryption and authentication

	CodeBlue	RehabSPOT	MAPS	TITAN	SPINE
Programming Effectiveness					
Programming Abstractions	partial support	partial support	✓	✓	✓
Software Engineering Methods		partial support	partial support	partial support	✓
Debugging and Testing Tools		✓	✓		✓
System Efficiency					
Resource Management Optimization	✓			✓	✓
System Interoperability					
Application-Level Communication Protocol	✓	✓	✓	partial support	✓
Adapter for Heterogeneous Sensor Inclusion					✓
System Usability					
GUI-based flexible management of the BSN system	partial support	partial support			✓
PC and smartphone-based Coordinator	partial support	✓	partial support	✓	✓
Privacy Support					
Data encryption and authentication		✓		✓	✓

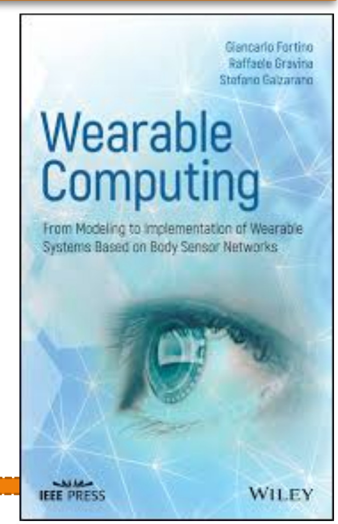
The SPINE (Signal Processing In-Node Environment) Project



http://spine.deis.unical.it

Project Contributors:

- Telecom Italia (F. Bellifemine)
- UT Dallas (R. Jafari)
- UC Berkeley (A. Sangiovanni-Vincentelli)
- ...



- G. Fortino, R. Giannantonio, R. Gravina, P. Kuryloski, R. Jafari, "Enabling Effective Programming and Flexible Management of Efficient Body Sensor Network Applications", IEEE Transactions on Human-Machine Systems, vol. 43, no. 1, pp. 115-133, Jan. 2013.
***A**ward: 2014 Andrew P. Sage Best IEEE SMC Transactions*
(from Web of Science Core Collection) Highly Cited Paper

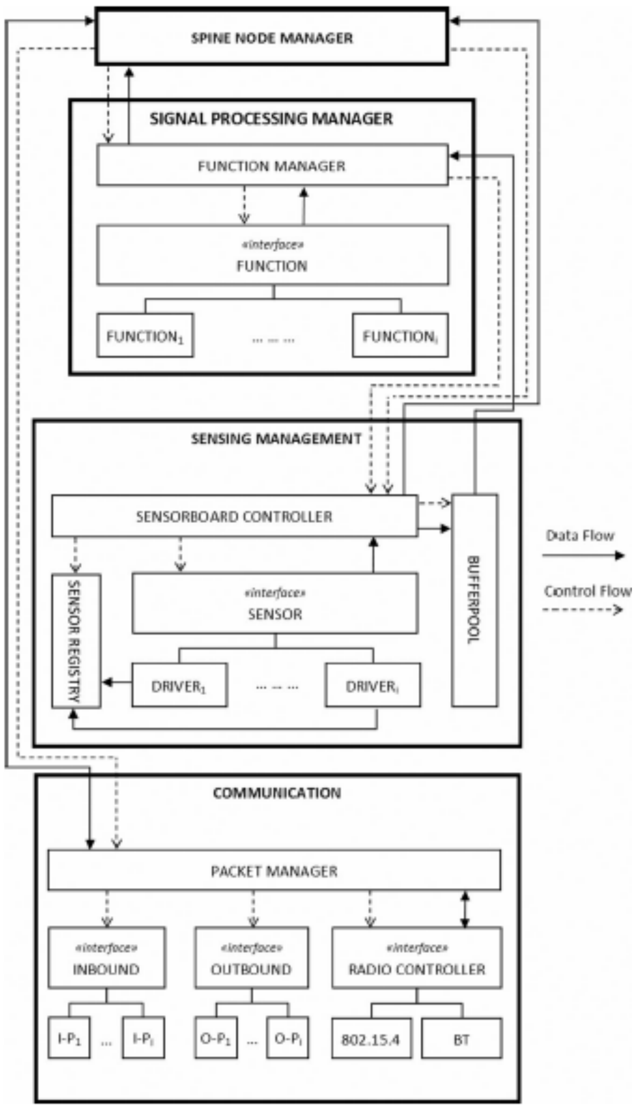
-G. Fortino, S. Galzarano, R. Gravina, *Wearable Systems and Body Sensor Networks: from modeling to implementation*, Wiley, USA, 2018.

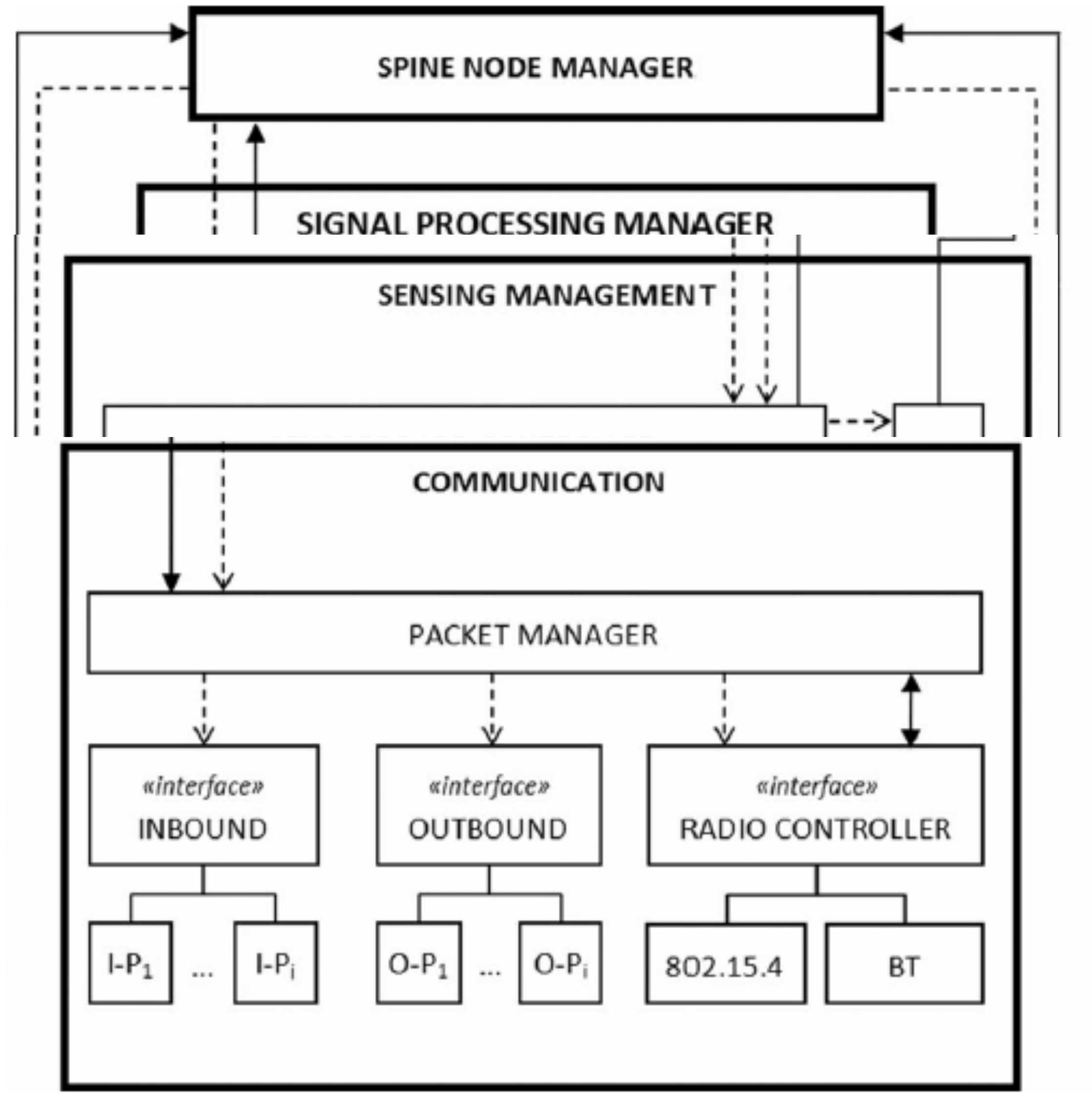
The SPINE (Signal Processing In-Node Environment) Framework

FUNCTIONALITY	DESCRIPTION
SERVIC	d their
FLEXIB	sensor
RAW S	mission
ON-NC	iodic
HIGH-I	and data ta.
TAILOF (*Exter*	and
SECUR	

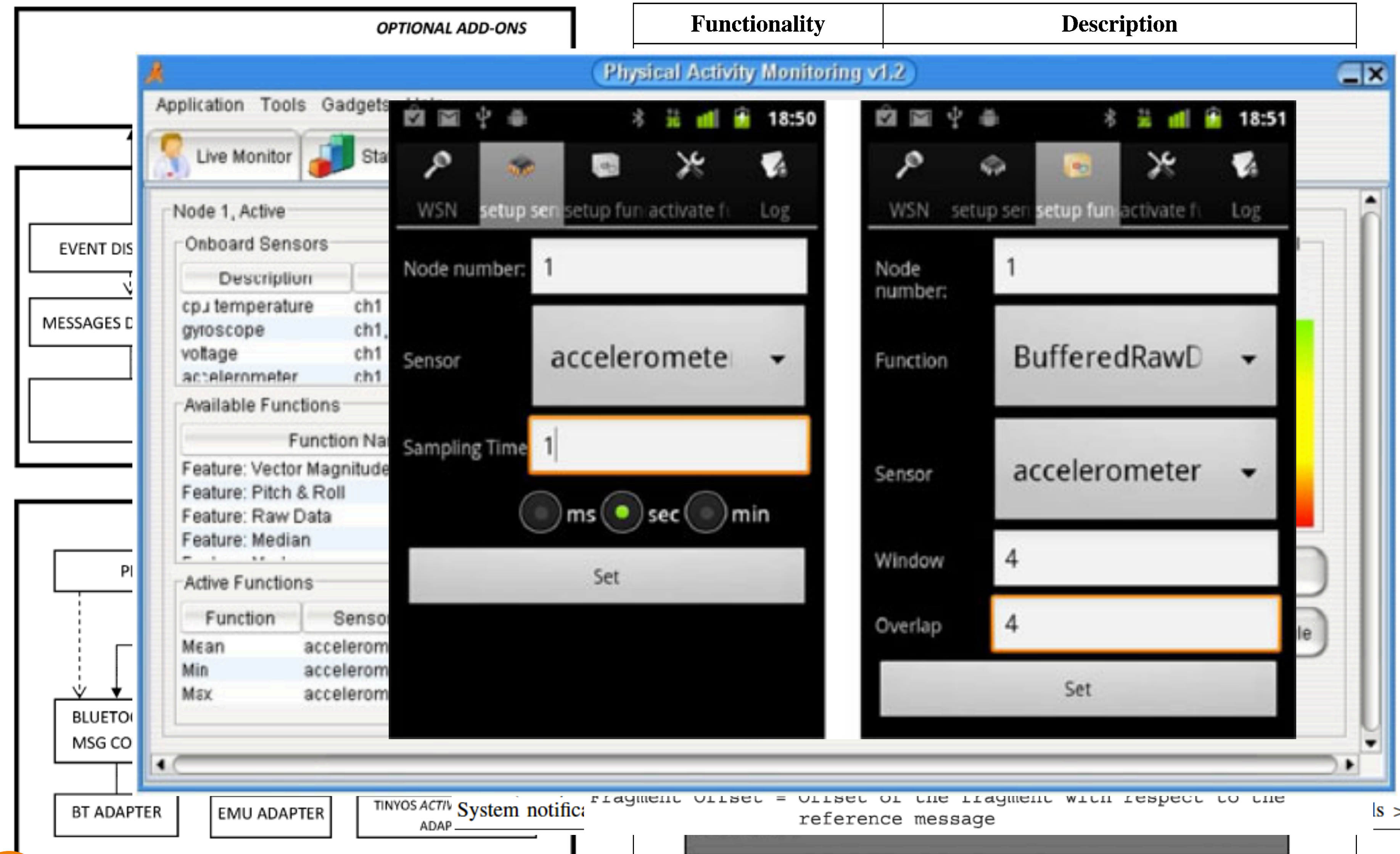
FEATURE	DESCRIPTION
Raw data	Sensed data coming from sensing processes
Max	Maximum value computed on a sample window
Min	Minimum value computed on a sample window
Range	Maximum displacement (max–min) value computed on a sample window
Mean	Average value computed on a sample window
Amplitude	(Maximum-mean) value computed on a sample window
RMS	RMS value computed on a sample window
St dev	Standard deviation value computed on a sample window
Total energy	Cross-axial magnitude computed on a sample window. It takes into account multiple sensor channels, if any.
Variance	Variance value computed on a sample window
Mode	Most frequent value computed on a sample window
Median	Median value computed on a sample window (central value of the ordered window buffer)
Vector magnitude	Magnitude of a sample window (sum of the squares of the window elements)
Entropy	Entropy computed on a sample window
Pitch & roll	Pitch and roll estimation computed on a sample window. It is useful only if applied to accelerometer data.
ALARM	
ABOVE	An alarm is triggered when a given sensor data or a computed feature exceeds the specified threshold
BELOW	An alarm is triggered when a given sensor data or a computed feature goes below the specified threshold
WITHIN	An alarm is triggered when a given sensor data or a computed feature are within the range of the specified thresholds (min, max)
OUTSIDE	Alarm is triggered when a given sensor data or a computed feature exceeds the range specified by the thresholds (min, max)

The SPINE Node architecture

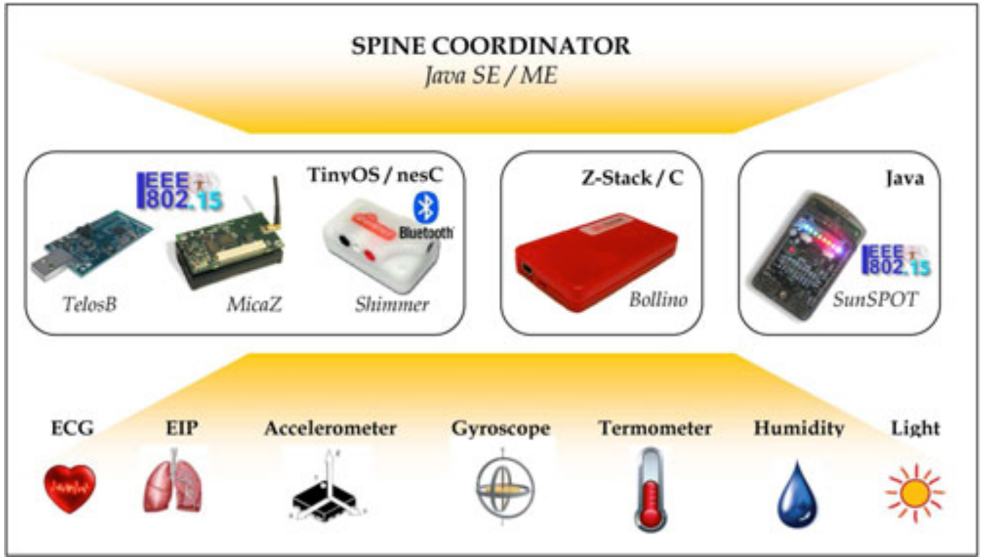




The SPINE Coordinator architecture



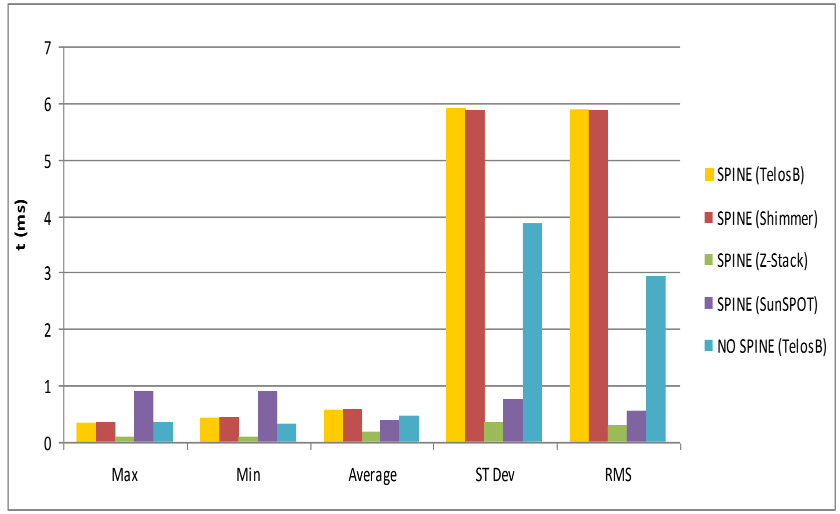
SPINE: Sensor and Coordinator Heterogeneity



SPINE-TESTED MOBILE PERSONAL DEVICES

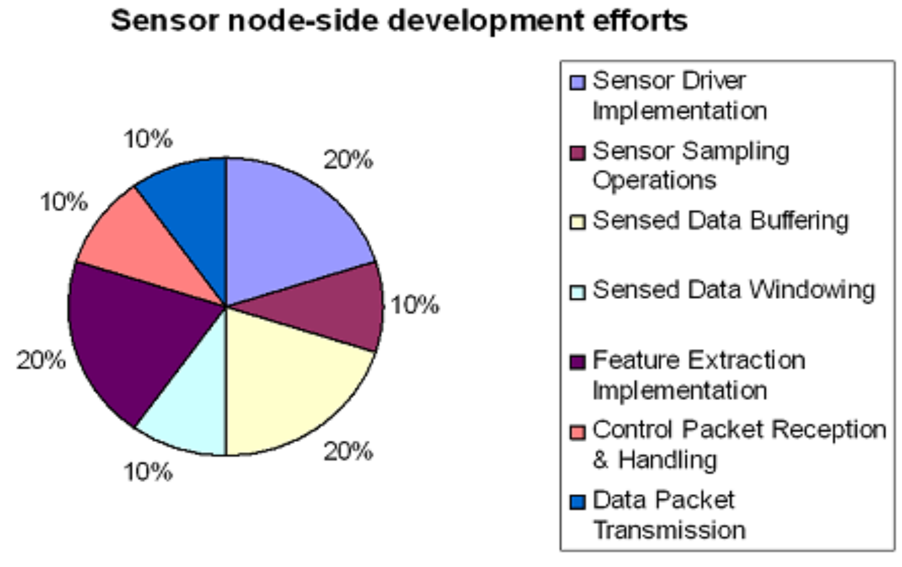
Device	CPU	RAM	Misc.
HTC NexusOne	1GHz, Qualcomm Snapdragon QSD 8250	512 MB	Android 2.x. MicroSD, up to 32 GB.
Samsung Galaxy S	1GHz, ARM Cortex-A8	512 MB	Android 2.x. MicroSD, up to 32 GB.
Nokia N95	332 MHz, TI OMAP 2420 (ARM11-based)	128 MB	Symbian OS v9.2, S60 rel. 3. MicroSD, up to 32GB
Nokia 6120	369, MHz ARM11	64 MB	Symbian OS v9.2, S60 rel. 3.1. MicroSD, up to 8GB.

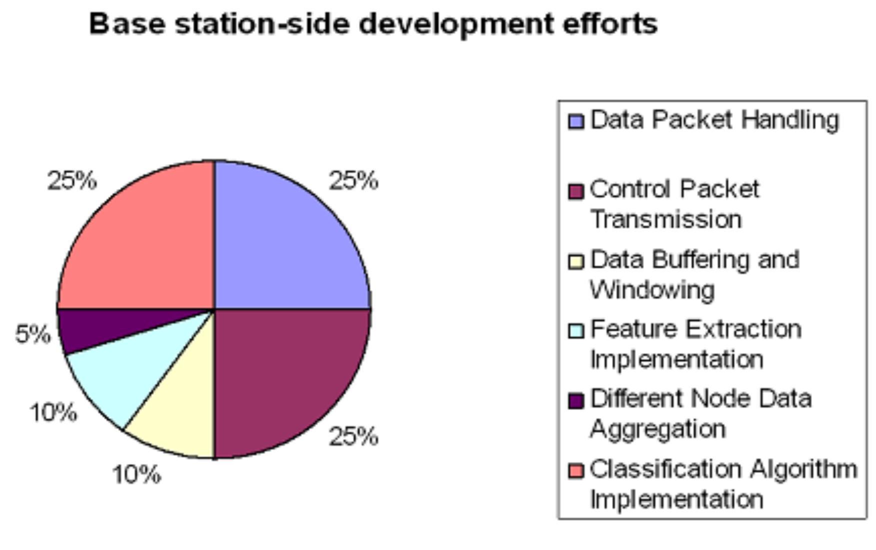
SPINE: Performance Evaluation



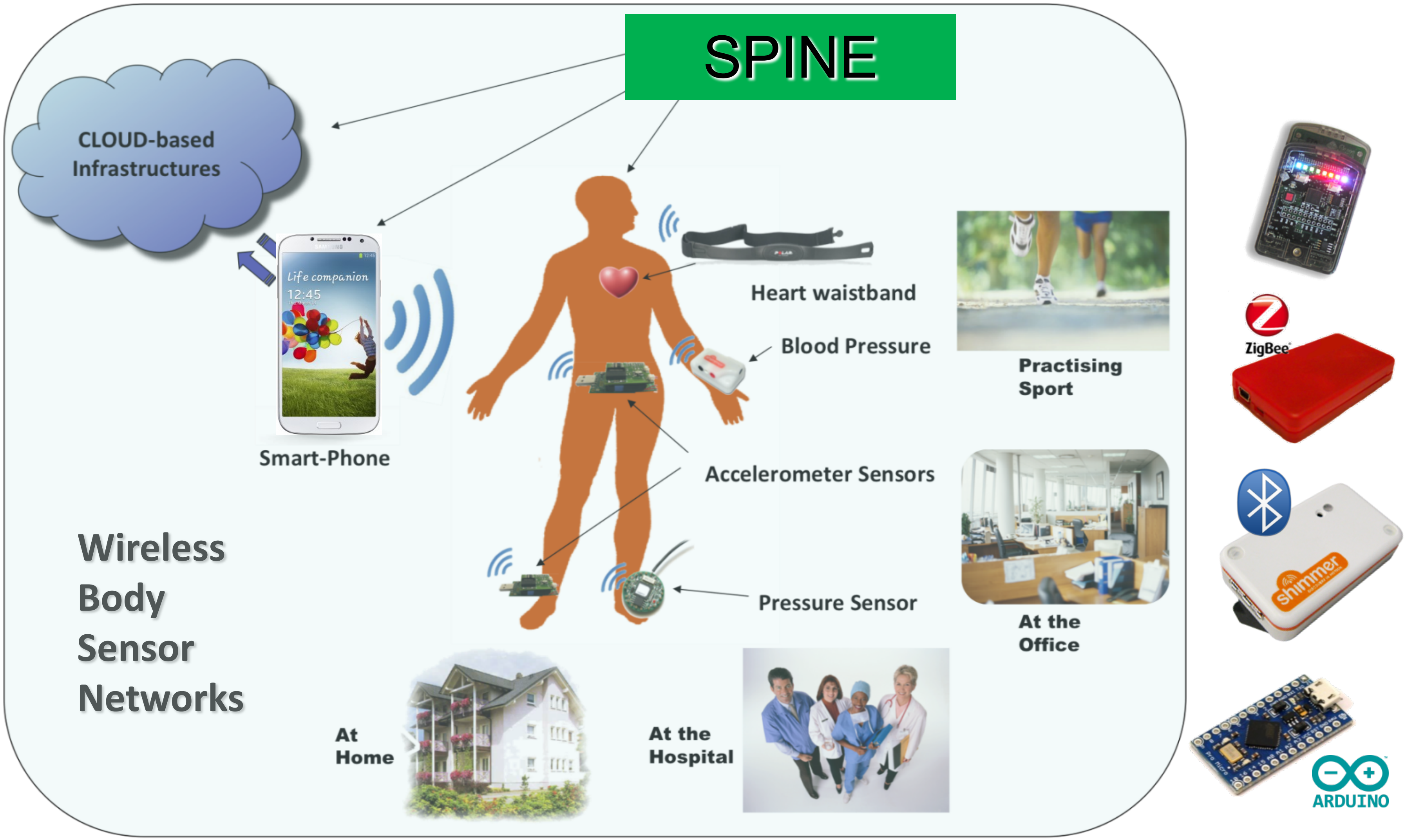
	Memory Requirements		*Energy Consumption*			*Bandwidth*	*Transmission Delay*	
Application Profile	**RAM (Kb used/av.)**	**ROM (Kb used/av.)**	**Average Power Consumption**	**Battery**	**Lifetime**	**Bitrate**	**802.15.4**	**Bluetooth**
SPINE on *TelosB*	3.7 / 10	33.5 / 48	6.6 mW/s	650mAh	101 h	178 byte/s	10,07	N/A
SPINE on *Shimmer*	4.4 / 10	40.0 / 48	13.9 mW/s	280mAh	21 h	178 byte/s	10,04	N/A
SPINE on *Shimmer Bluetooth*	4.3 / 10	34.4 / 48	87.8 mW/s	280mAh	3 h	150 byte/s	N/A	3,05
SPINE on *Z-Stack*	3.9 / 8	95.9 / 128	11.2 mW/s	650mAh	60 h	160 byte/s	0,61	N/A
SPINE on *SunSPOT*	79.0 / 512	75.0 / 4096	84.2 mW/s	720mAh	9 h	168 byte/s	67,20	N/A
A-hoc application on *TelosB*	1.3 / 10	16.1 / 48	73.7 mW/s	650mAh	9 h	152 byte/s	9,98	N/A
CodeBlue on *TelosB*	3.4 / 10	36.6 / 48	74.2 mW/s	650mAh	9 h	186 byte/s	10,25	N/A
TITAN on *TelosB*	9.0 / 10	38.7 / 48	18.7 mW/s	650mAh	36 h	158 byte/s	10,43	N/A

SPINE: Evaluation of Development Efforts





Application Domains



SPINE-based Research BSN Prototypes

- **Human Activity Recognition**
  - higher accuracy using less wearable nodes compared to related work
- **Step-counter**
  - Works for elderly using walkers, and for disabled people
- **Emotional Stress indicator**
  - Real-time, evaluation based on time-domain analysis
- **Physical Energy Expenditure**
  - Auto-orientation for gravity-compensation

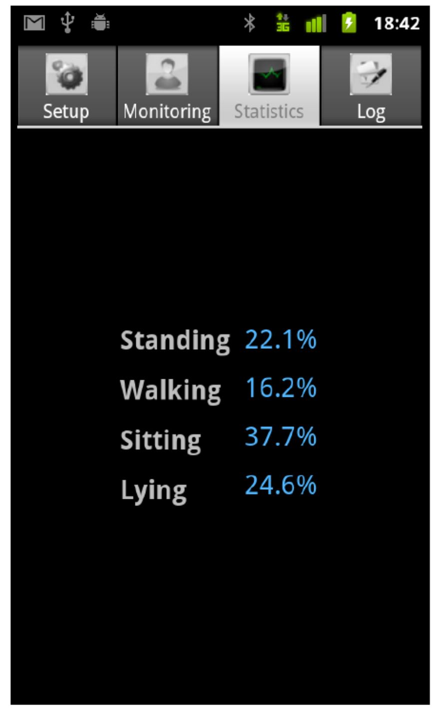
Research BSN Prototypes

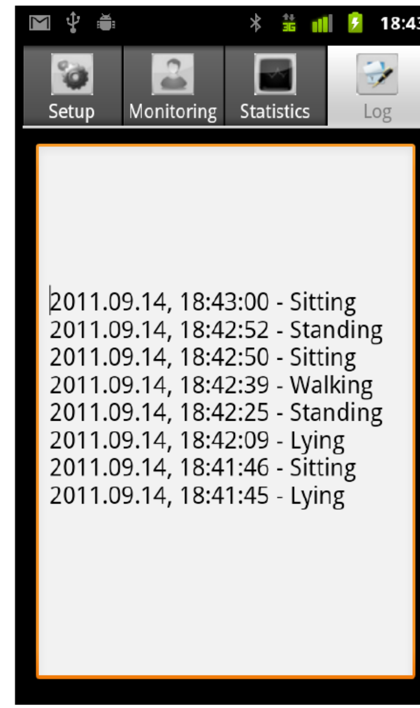
Application	SPINE Node-side	SPINE Coordinator-side
Activity Recognition	None	Off-line feature selection algorithm. Real-time K-Nearest Neighbor classifier.
Physical Rehab	None	Trigonometric function for upper and lower body joint angles estimation.
Gait Analysis	HMM on-line classifier (trained off-line using Matlab). Custom-defined gait data message encoder.	Custom-defined gait data message decoder.
Kcal Expenditure	Activity counts feature extraction.	Kcal estimation algorithm, based on activity counts values from wearable node.
Stress Detector	Driver adapter for Cardio-Shield sensor-board. RRi feature extraction.	Time-domain HRV analysis. Stress detector classifier.
Handshake Detection	Zero-crossing and Near-zero feature extraction. On-line local decision tree classifier for potential handshake detection.	Extension for Collaborative-BSN interaction model. Distributed decision tree classifier based on joint data from potential meeting people.

Human Activity Recognition

- System configuration: 2 wireless motion sensors (waist and thigh) + 1 coordinator device
- Recognized Activities and posture: walking, standing, sitting, lying
- Built-in **Fall Detection**
- Step-counter
- Energy Expenditure (Kcal)
- Very Accurate (>98%)
- **Connection to the Cloud**

Human Activity Recognition



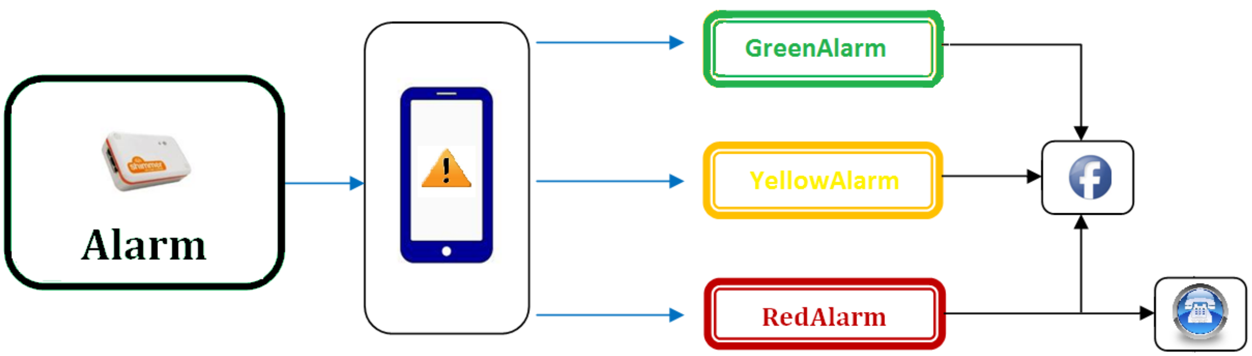


Play VIDEO

Smart Fall Detection

- System configuration: 1 wireless motion sensor + 1 coordinator device
- Better precision, specificity, and sensitivity than other commercial systems
- Alarm notification on **Social Networks** and via **automated SMS/voice calls**

$$\frac{m\ activities}{n\ activities} = LYING$$



Smart Fall Detection

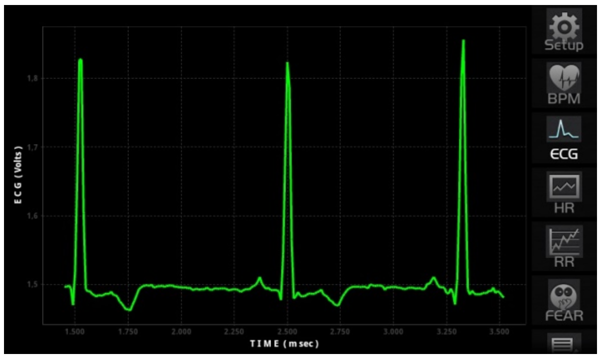
- Wide training set including falls and daily activities

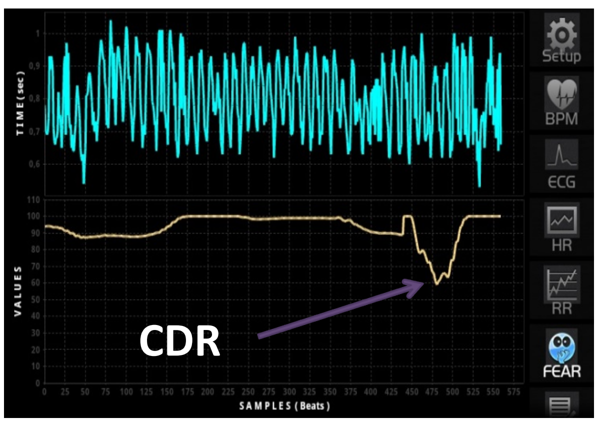
- **Falls**
  - Forward
  - Backward
  - Side
  - Faint
- **Daily Activities**
  - Lying down
  - Walking
  - Running
  - Sitting
  - ...

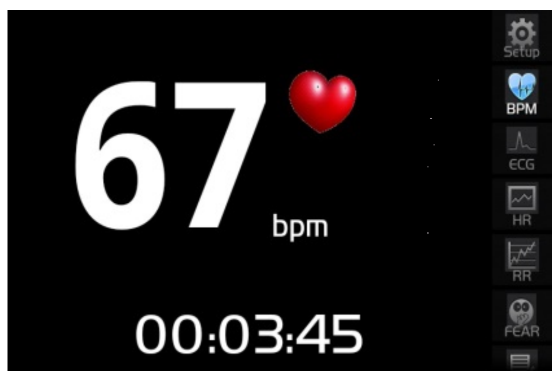
Cardiac Monitoring, Mental Stress Detection and Fear Recognition

- Real-time, evaluation based on time-domain analysis
- Low power
- Completely wireless and non-invasive
- Wearable
- Robust to motion artifacts
- **Connection to the Cloud**

Cardiac Monitoring, Mental Stress Detection and Fear Recognition







DEMO

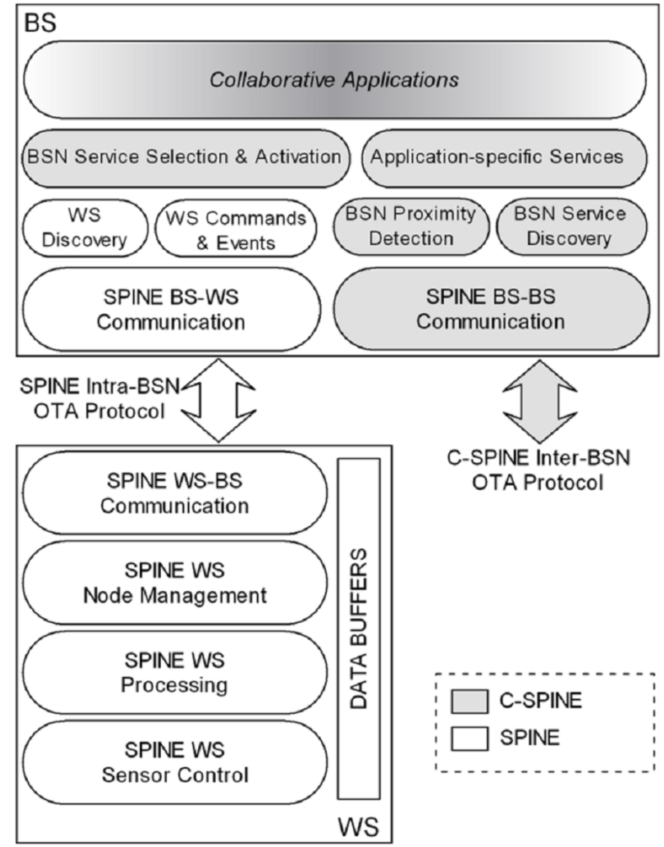
SPINE Enhancements and Variants

- C-SPINE
- Virtual Sensors and SPINE
- A-SPINE
- SPINE2
- SPINE-*

SPINE Enhancements

- **C-SPINE**
  - Support for **Collaborative BSNs**

*Giancarlo Fortino, Stefano Galzarano, Raffaele Gravina, Wenfeng Li: A framework for collaborative computing and multi-sensor data fusion in body sensor networks. Information Fusion 22: 50-70 (2015)



- **Virtual Sensors and SPINE**
  - **Multi-layer task model** based on the concept of VS
  - *Definition*:

$$VS_i = \{I_i, O_i, C_i\}\text{, where } C_i = \{\vec{t}_{in}, \vec{t}_{out}, d, p\}$$

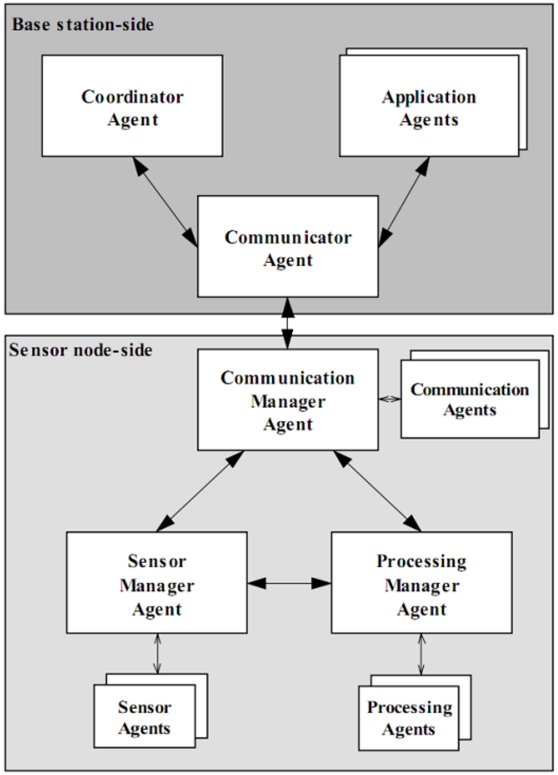
  - VSs are composable to form more complex VSs

*N. Raveendranathan, S. Galzarano, V. Loseu, R. Gravina, R. Giannantonio, M. Sgroi, R. Jafari, G. Fortino, “From Modeling to Implementation of Virtual Sensors in Body Sensor Networks”, IEEE Sensors Journal, 12(3), 583-593, 2012.

A-SPINE: Agent-based programming for BSNs

- Among the programming paradigms proposed for the development of WSN applications, the **(mobile) agent-based paradigm** can effectively deal with the programming issues that WSNs have posed.
- MAPS Case study:

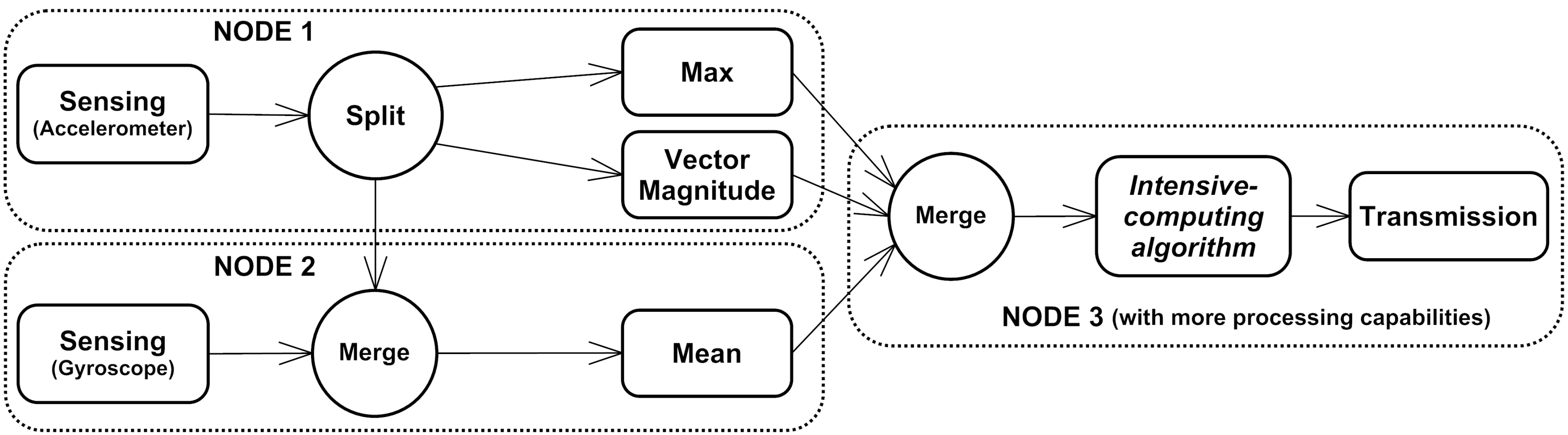
  Physical activity monitoring application based on SunSPOT sensor nodes.



- F. Aiello, G. Fortino, R. Gravina, A. Guerrieri, "A Java-based Agent Platform for Programming Wireless Sensor Networks" *The Computer Journal*, 54(3), pp.439-454, 2011.
- F. Aiello, F. Bellifemine,S. Galzarano, R. Gravina, and G. Fortino "An agent-based signal processing in-node environment for real-time human activity monitoring based on wireless body sensor networks", *Journal of Engineering Applications of Artificial Intelligence. Vol. 24, n. 7, pp. 1147-1161.* 2011.

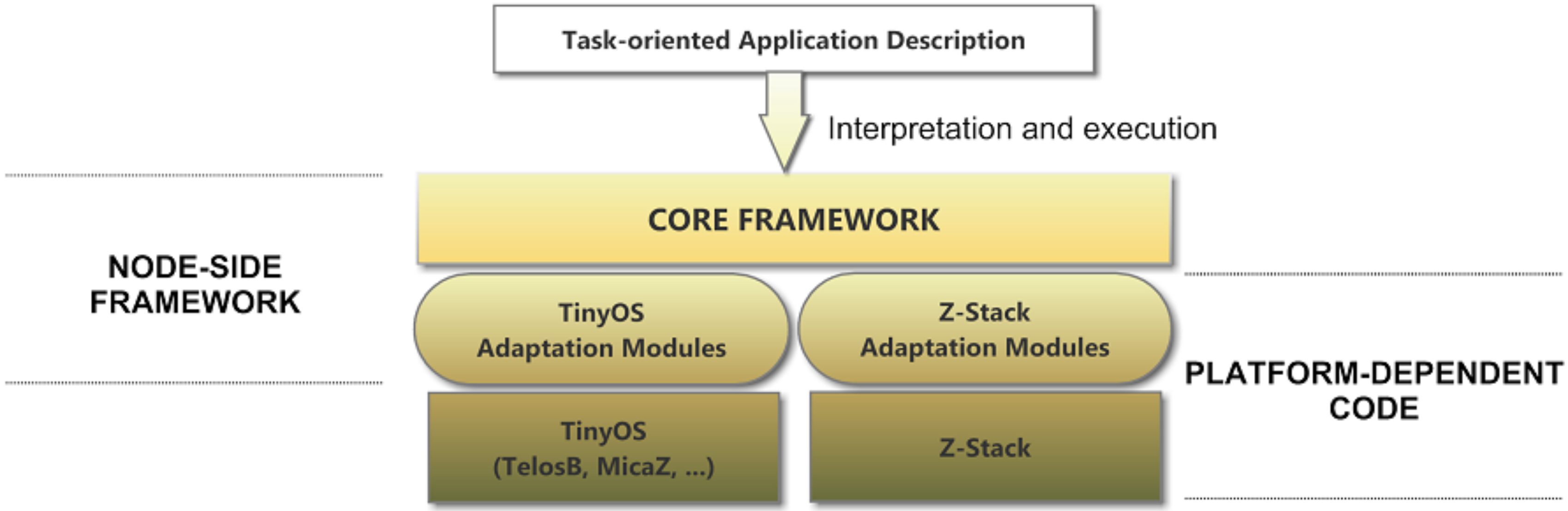
The SPINE2 framework: Task-oriented programming of WBSNs

Signal Processing In Node Environment

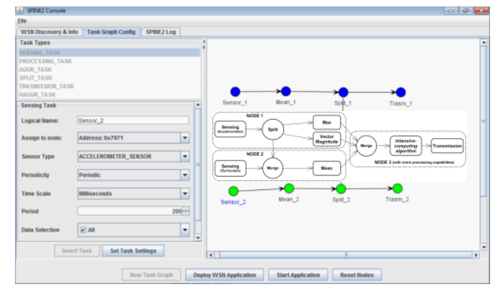


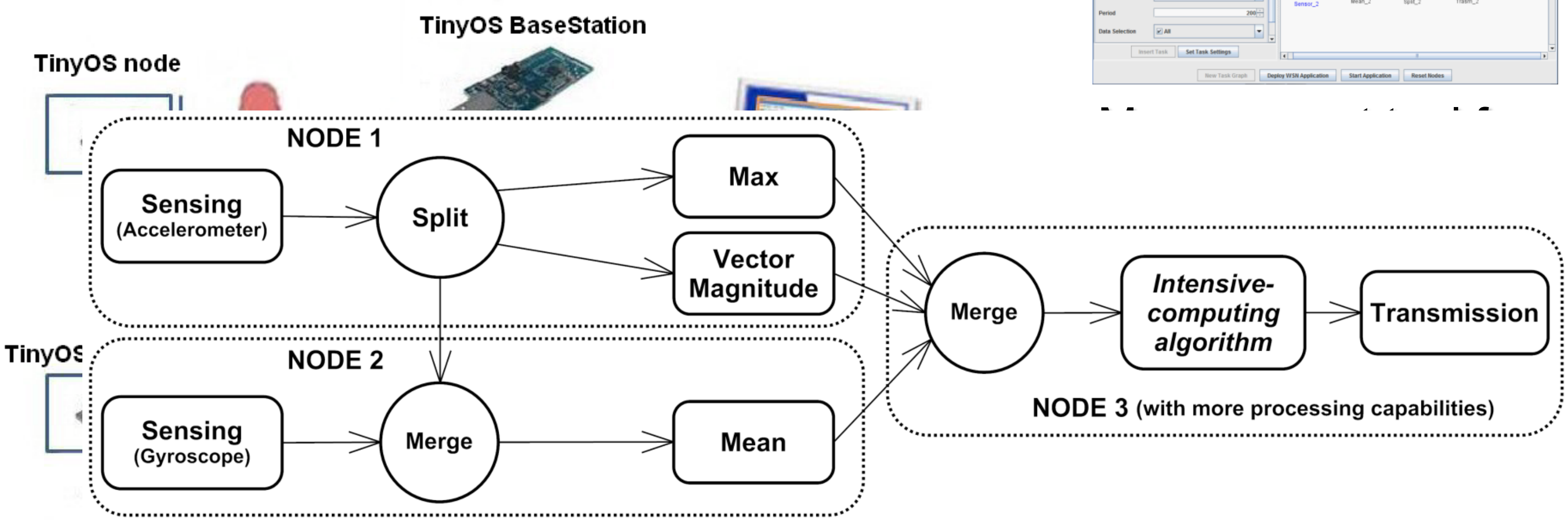
S. Galzarano, R. Giannantonio, A. Liotta, G. Fortino, "A task-oriented framework for networked wearable computing", in *IEEE Transactions on Automation Science and Engineering*, 13(2): 621-638 (2016).

The SPINE2 framework



SPINE2: developing BSN applications

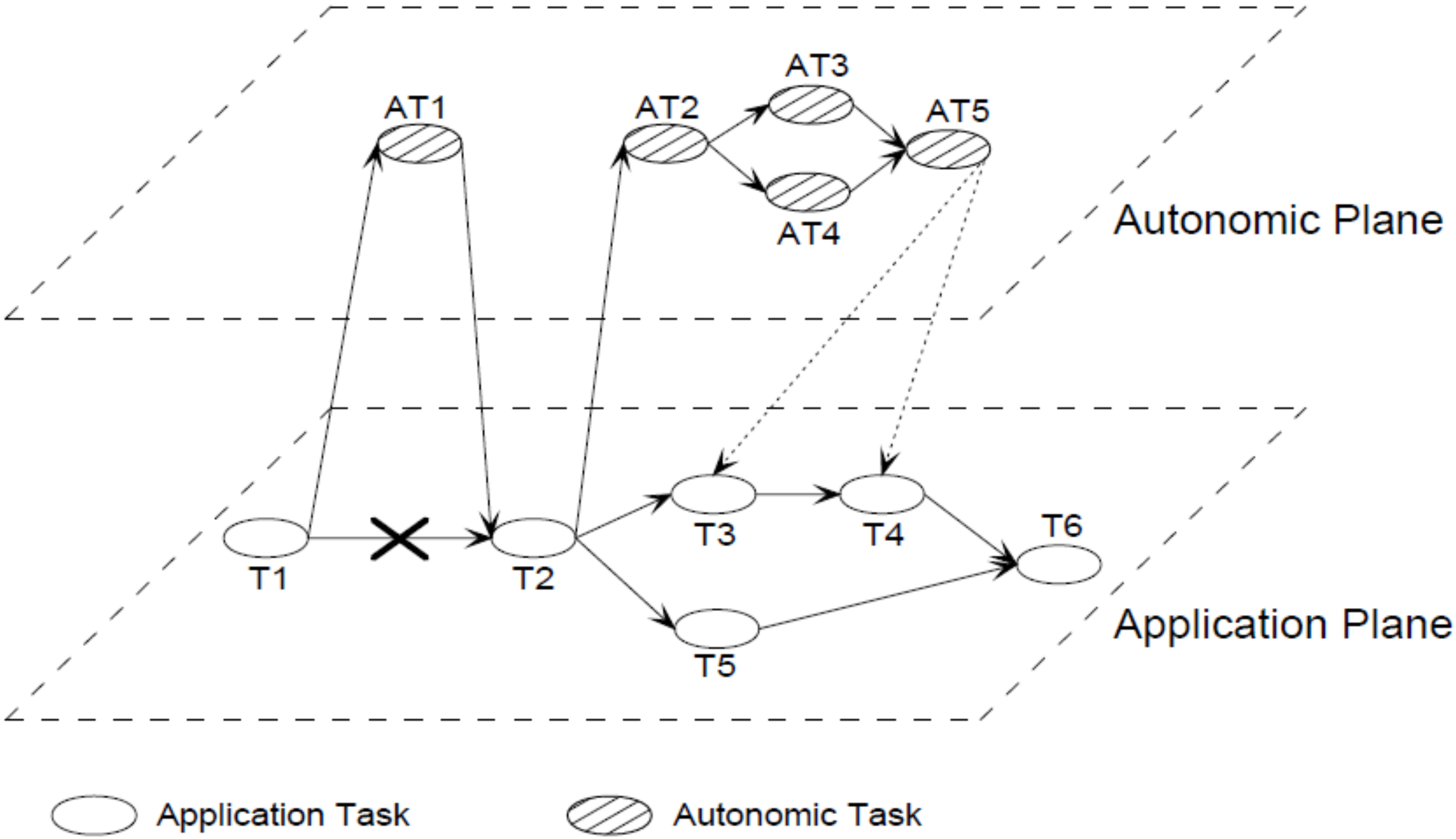




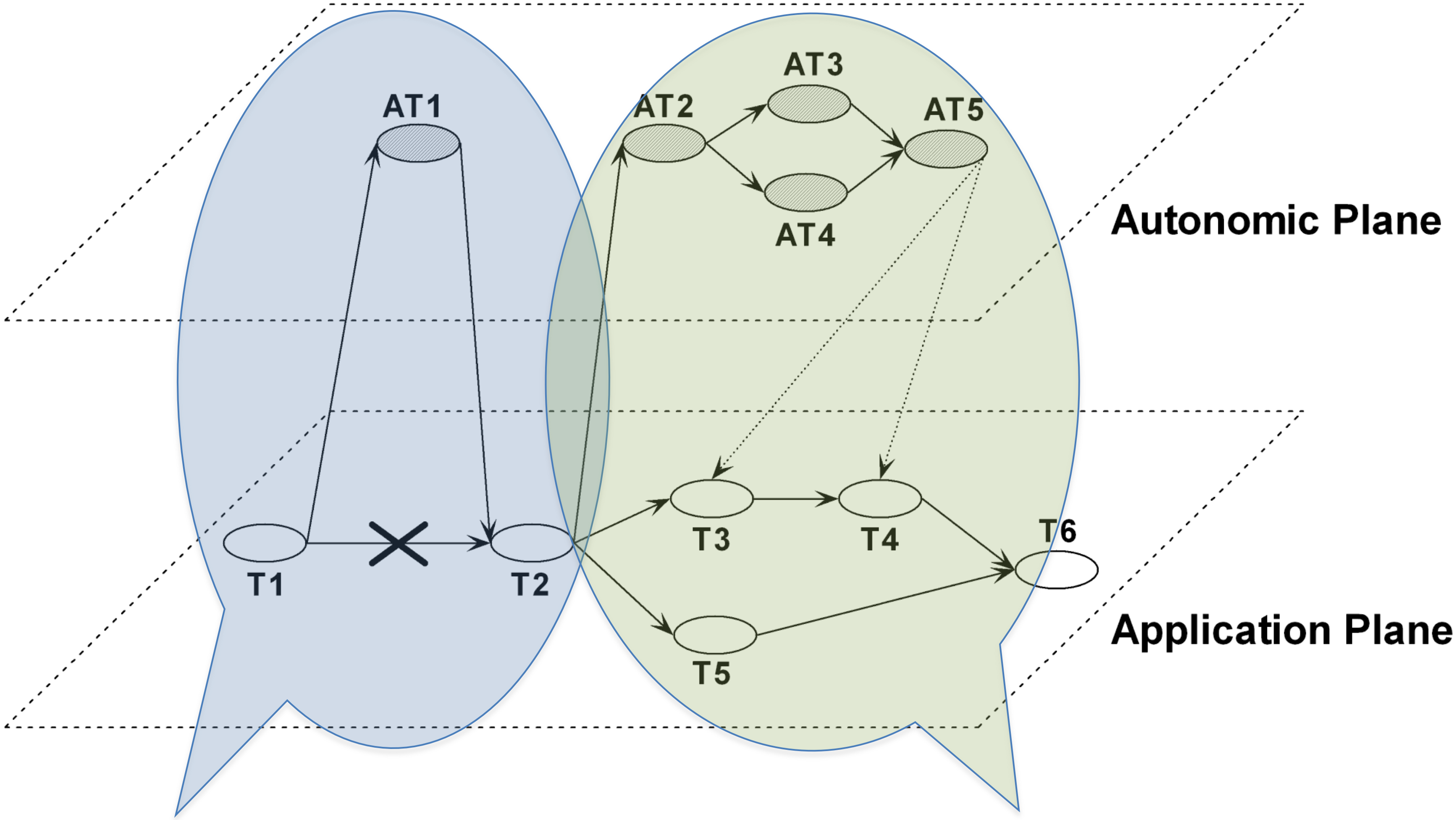
SPINE-* autonomic architecture

- ***Autonomic*** characteristics may be incorporated into the system, so to better ensure the reliability and maintainability properties.

- The architecture is composed of two distinct planes



SPINE-* autonomic architecture

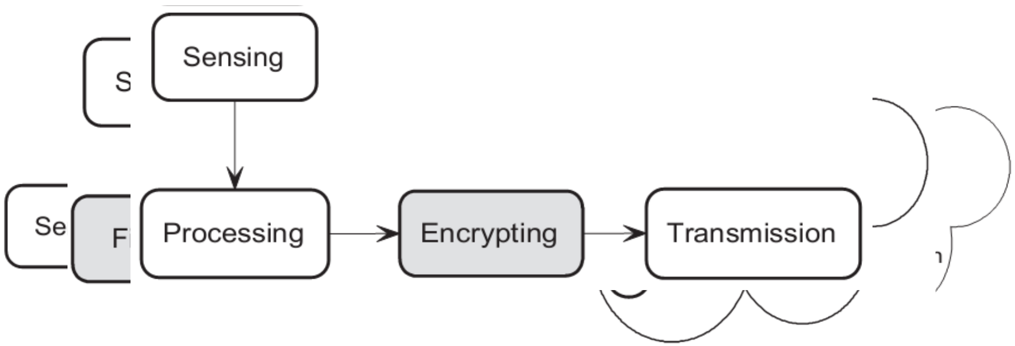


Case 1: Bypass
(Extra data processing)

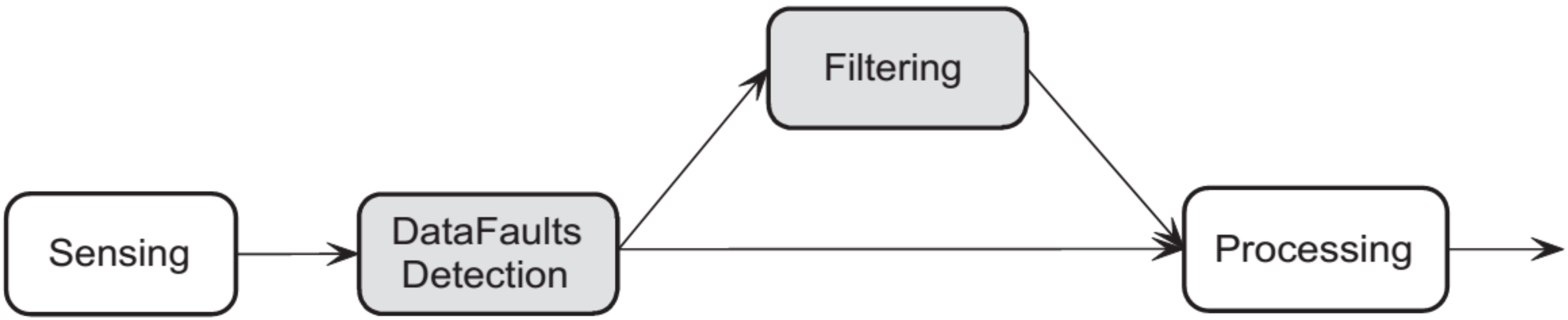
Case 1: Feedback loop
(Run-time reconfiguration)

Self-* properties

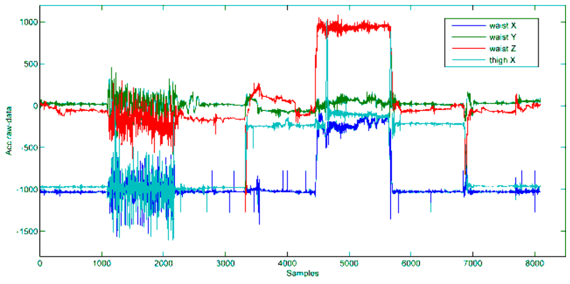
- Self- configuration
- Self- healing
- Self-optimization
- Self-protection



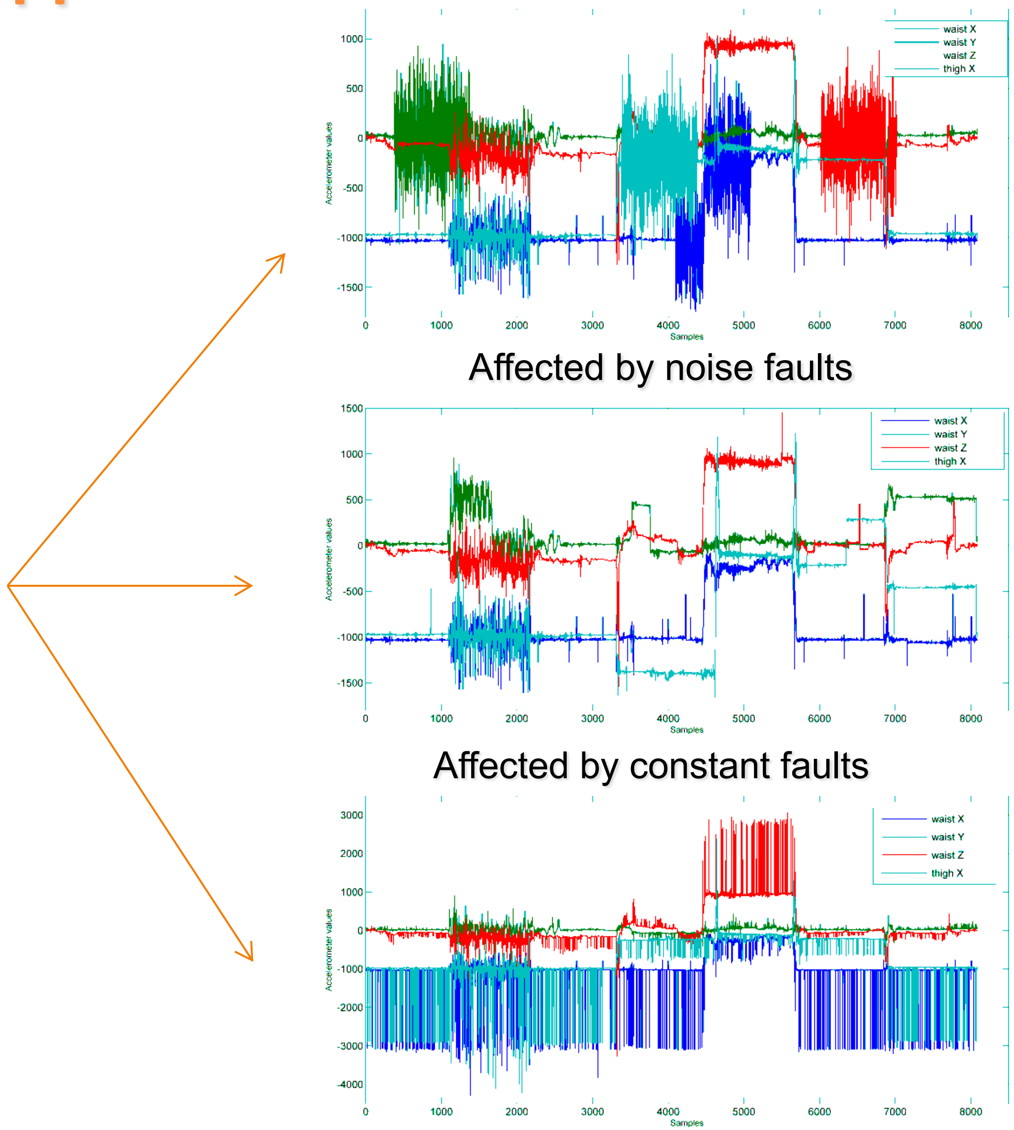
A *self-healing* BSN application



A *self-healing* BSN application



Original accelerometer raw-data streams.

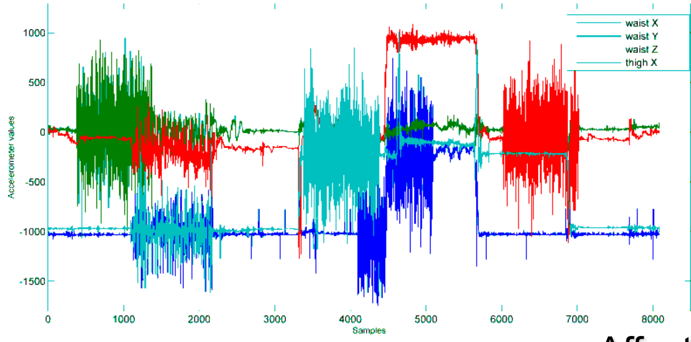


Affected by noise faults

Affected by constant faults

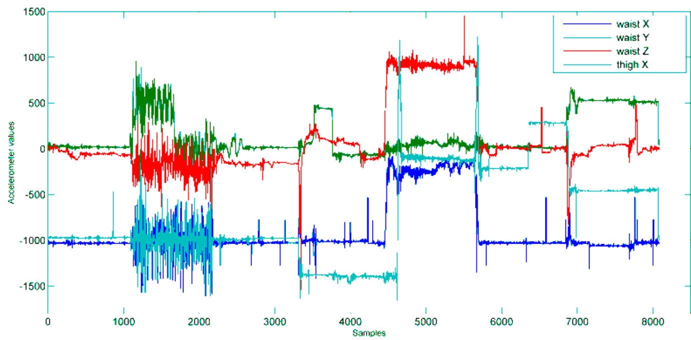
Affected by short faults

A *self-healing* BSN application



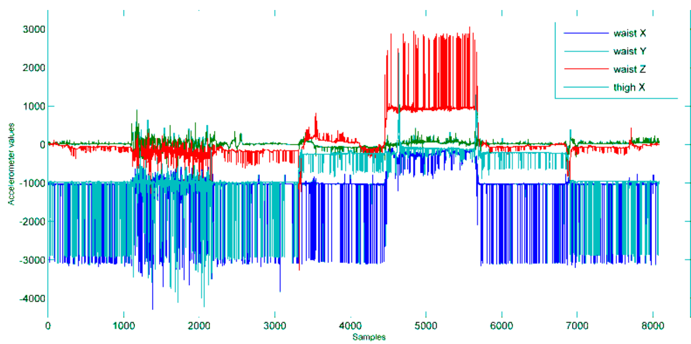
Channel	σ	Accuracy
All	100	98.64%
All	300	91.77%
All	500	89.54%
All	1000	88.29%
All	1500	84.85%

Affected by noise faults



Channel	K	Accuracy
All	100	98.51%
All	200	95.24%
All	500	91.06%
All	1000	82.08%
All	2000	56.65%

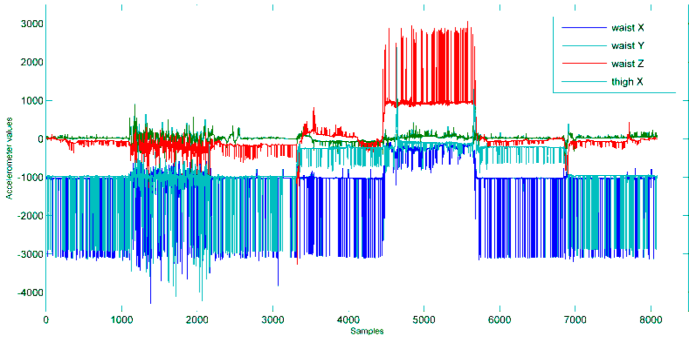
Affected by constant faults



Channel	P	Accuracy
All	1%	79.90%
All	5%	55.09%
All	10%	51.86%
All	25%	48.14%
All	50%	46.65%

Affected by short faults

A *self-healing* BSN application



Channel	P	Accuracy
All	1%	79.90%
All	5%	55.09%
All	10%	51.86%
All	25%	48.14%
All	50%	46.65%

Affected by short faults

ACCURACY IMPROVEMENTS WITH SHORT-FAULTS OVER ALL CHANNELS AND WITH *C=3*

P	Accuracy (affected data)	Accuracy (recovered data)
1%	79.90%	99.75%
5%	55.09%	99.75%
10%	51.86%	98.51%
25%	48.14%	59.55%
50%	46.65%	47.64%

Platform-Based Design methodology for BSN

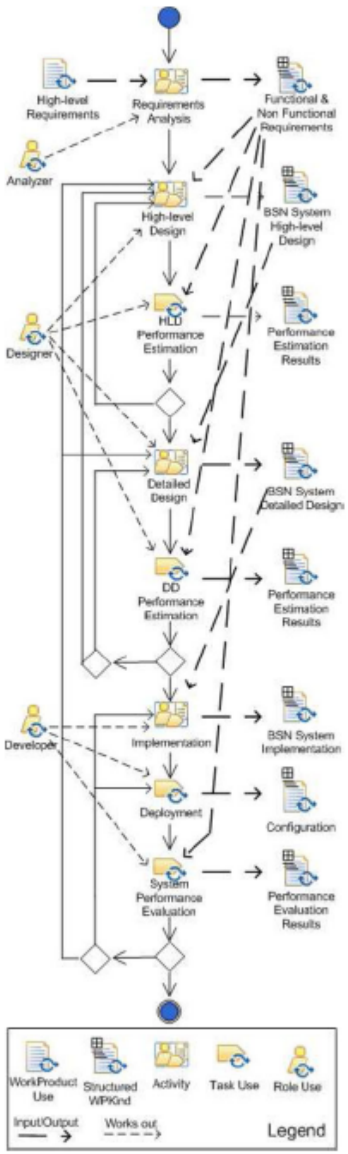
- PBD is a methodology for system-level design for embedded systems.
- It is based on stacks of ***Platforms*** representing the system at different level of abstractions.
- Iterative refinements map components of a platform to components of the lower platform, providing an ***implementation*** of the higher platform.
- Each ***mapping*** step is obtained as the resolution on an optimization problem.
- At the end, the system is fully specified and a final, optimized and correct-by-construction implementation is available.

Platform-Based Design methodology for BSN

- The first layer is an application interface called **Sensor Network Service Platform**. The SNSP defines a set of services available to the end user to specify the target application formally without dealing with the details of a particular network implementation.
- The abstraction layer in the middle is the **Sensor Network Ad-hoc Protocol Platform** and it is a library of communication protocols that can be optimized to be deployed on the given topology to satisfy end to end communication constraints.
- The lowest abstraction layer is the **Sensor Network Implementation Platform**, which is a library of different hardware platforms that can be used to create the topology that supports the application.

- The approach has been experimented on a case study for complex **human activity recognition** based on **Template Matching** techniques

Platform-Based Design Methodology for BSN



SPEM 2.0 Process Schema:

- Requirements Analysis
- High-level Design (HLD)
- Performance Estimation of HLD
- Detailed Design (DD)
- Performance Estimation of DD
- Implementation
- Deployment
- System Performance Evaluation

Conclusions

- ***SPINE*** (and its enhancements and variants) is a significant step forward to the state-of-the-art:
  - Support for heterogeneous sensor platforms;
  - High-level abstractions (Virtual Sensors);
  - High-level Data Processing (support for signal processing and classification);
  - Platform-independence;
  - Novel Interaction models (support for Collaborative BSNs).

- Each proposed BSN **prototype** improved the state-of-the-art.
- ***PBD*** methodology has never been applied to the BSN domain before.

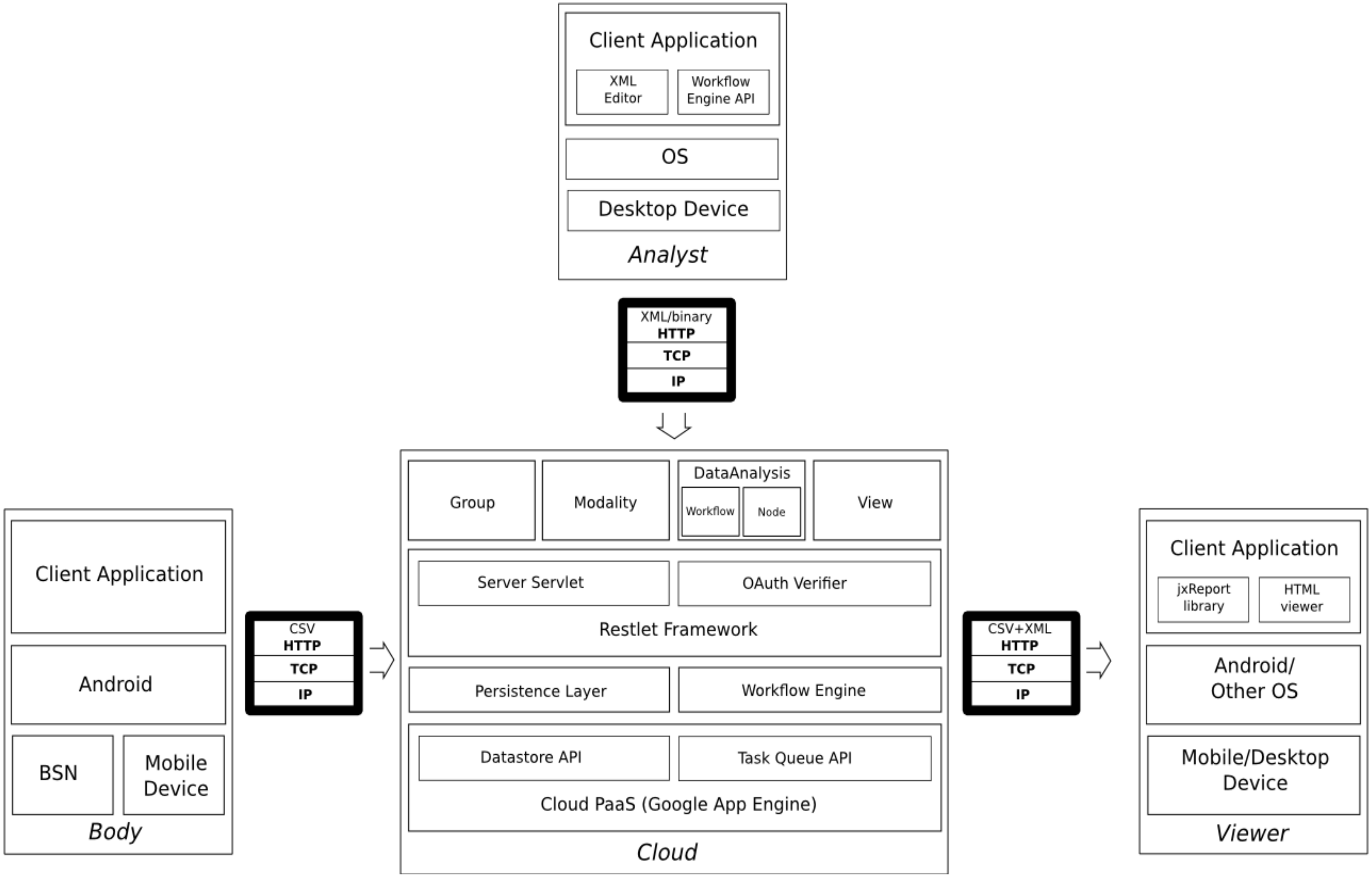
Research Challenges: Towards BSN communities

- ***BSNs*** can be adopted to monitor a large pool of people, thus generating large amounts of contextual data. Such a Big Data scenario requires a scalable approach for data collection, storage, processing and analysis. Cloud computing can provide a flexible storage and processing infrastructure to perform both online and offline analysis of data streams generated by BSNs.

- We have proposed ***BodyCloud (bodycloud.dimes.unical.it)***, a Cloud-enabled SaaS architecture for the management of body sensor data streams and the complete life cycle of data analysis workflows (data collection, storing, analysis, and presentation).

- ***BodyCloud*** approach offers a very flexible and intuitive programming model centered on a few web-based programming abstractions (group, modality, workflow/node, view) that allow to define and deploy community BSN applications.

The BodyCloud Architecture



G. Fortino, D. Parisi, V. Pirrone, G. Di Fatta, BodyCloud: A SaaS Approach for Community Body Sensor Networks, *Future Generation Computer Systems*, vol. 35, n. 6, pp. 62-79, 2014.

Among the Most Cited Future Generation Computer Systems Articles - The most cited articles published since 2012, extracted from Scopus: https://www.journals.elsevier.com/future-generation-computer-systems/most-cited-articles

Future Work (about WBSN/SPINE)

- Enhancing and Applying the SPINE* approach to make BSN applications ***Cognitive***
- Software-defined BANs
- Integration of BSNs and Environmental WSNs (e.g. building networks): gateway-based approach integrating SPINE and BMF (http://bmf.deis.unical.it)
- Integration of BSNs in IoT Smart Environments (e.g. smart building, smart office, etc.): ACOSO (http://acoso.dimes.unical.it)

Highlights on SPINE works

SPINE Open Source Project

http://spine.deis.unical.it

BODYCLOUD Open Source Project

http://bodycloud.dimes.unical.it

-G. Fortino, R. Giannantonio, R. Gravina, P. Kuryloski, R. Jafari, “Enabling Effective Programming and Flexible Management of Efficient Body Sensor Network Applications”, IEEE Transactions on Human-Machine Systems, vol. 43, no. 1, pp. 115-133, Jan. 2013.

Awarded with Andrew P. Sage Best SMC Transactions Paper 2014

N. Citations to SPINE-enabled research (overall) ≈ 4000

BOOK: G. Fortino, S. Galzarano, R. Gravina, *Wearable Systems and Body Sensor Networks: from modeling to implementation*, IEEE Press - Wiley, USA, 2018.

Call for Authors

Wiley-IEEE Press Book Series on

Human-Machine Systems

We are looking for book proposals covering the following areas:

- Affective Computing
- Assistive Technology
- Augmented Cognition
- Brain-based Information
- Human-Machine Communications
- Companion Technologies
- Design Methods
- Entertainment Engineering
- Human Factors
- Human Performance Modeling
- Human-Computer Interaction
- Human-Machine Cooperation and Systems
- Human-Machine Interface and Communications
- Information Systems for Design/Marketing
- Information Visualization
- Interactive Design Science and Engineering
- Interactive and Digital Media
- Kansei (sense/emotion) Engineering
- Medical Informatics
- Mental Models
- Multimedia Systems
- Multi-User Interaction
- Resilience Engineering
- Supervisory Control
- Systems Safety and Security
- Team Performance and Training Systems
- User Interface Design
- Virtual and Augmented Reality Systems
- Wearable Computing
- Web Intelligence and Interaction

Series Scope Statement

The IEEE Press Series on Human-Machine Systems, established in June 2019, is published by IEEE Press and Wiley and covers integrated human/machine systems at multiple scales, including areas such as human/machine interaction; cognitive ergonomics and engineering; assistive/companion technologies; human/machine system modeling, testing and evaluation; and fundamental issues of measurement and modeling of human-centered phenomena in engineered systems.

BOOK TYPES

The series includes authored, edited or handbooks. The series will not publish conference proceedings or post-proceedings.

BOOK PROPOSAL SUBMISSION

Prospective authors should contact the Series Editor at **g.fortino@unical.it,** in order to discuss their project. Authors will then be invited to submit a book proposal to be reviewed by a panel of experts.

IEEE PRESS

WILEY

THANKS FOR YOUR ATTENTION

Questions?