Self-Aware Fog Computing in Private and Secure Spheres

Kalle Tammemäe[1]*, Axel Jantsch[2], Alar Kuusik[1],
Jürgo-Sören Preden[3], and Enn Õunapuu[1]

[1] Tallinn University of Technology, Estonia
{kalle.tammemae,alar.kuusik,enn.ounapuu}@ttu.ee
[2] Vienna University of Technology, Austria
axel.jantsch@tuwien.ac.at
[3] Thinnect, Inc.
jurgo@thinnect.com

Abstract. In real-time health analytics, smart cities, military sensing systems and others, big data analytics is enabled by the introduction of appropriate sensing and actuation systems. The introduction of next generation of sensing and actuation systems or the Internet of Things era have been facilitated by affordable low-power 32-bit microcontrollers combined with low-cost and effective sensors with appropriate power supplies, mobile and local data collection (local big data) capabilities, adaptive behavior using machine learning and evolving model-based behavior, etc. While Cloud computing offers big data processing and actuation capability at the server level, mist computing offers data processing and actuation capability at the very edge of the network. Fog computing offers the same capability in the middle at edge gateways. Mist computing is an enabler for many applications, which cannot be realized with alternative methods, such as smart cities, where city streets adapt to the changes happening in the city, socially intelligent houses where indoor environment management is integrated with inhabitants health monitoring or military sensing systems where situational information is automatically deduced from raw data and delivered to the information consumers. While these visionary applications promise to change our environment and the way we interact with the environment we face serious challenges in the implementing these systems, such as reliability of data exchange between nodes and routers, power distribution, quality of decision making etc.

1 Introduction

As our capacity for monitoring and measuring objects, activities and processes is growing exponentially, we also find many new applications in our immediate environments, our bodies and our homes. Wearable sensors for measuring our leisure and sports activities as well as our

* corresponding author

health conditions have proliferated and gained acceptance. Already today many of us continuously carry around several sensing, computing and communicating devices wherever we go. Also, our homes are becoming increasingly smart as they are equipped with cameras, motion detectors, and environmental sensors that provide data for air conditioning controllers, surveillance, and medical monitoring. The advances of sensing, computation and communication technology are also being utilized in military applications. While the military solutions used to represent the cutting edge of technology, with the rapid advancement of computing and sensing technology civilian applications are showing the way for future solutions.

The focal topic of the book - where and how the data processing should be performed in future systems, is a very relevant one for the private spaces of our bodies and our homes due to three main considerations: efficiency, privacy and dependability. The alternatives to be considered are between the two extremes of fully centralized and fully distributed processing. All the sensed data can be sent to a cloud server, recorded, archived, processed and used for making decisions that are then either returned to an actuator close to the sensor, or sent to another appropriate agent like a hospital. At the other end of the scale the sensor nodes themselves could do almost all the data analysis and decision making and only selected, abstracted data is sent to outside agents such as the hospital which is necessary to realize a required function.

Transferring data to the cloud for processing takes time and resources, and the delays are not always acceptable in latency sensitive applications like health-monitoring, emergency-response etc. [1]. The solution is to introduce a hierarchy where time-sensitive processing is done at the edge of the network. This is where Fog computing emerged along with IoT, with the role to mediate resource-abundant and slow cloud computing and agile but only partially informed edge computing. Edge computing can offer especially low latency response by providing limited computing resources at the very edge of the network. A more capable variant of Edge Computing - "Mist computing" offers the same benefits with more flexibility and manageability - with Mist computing nodes forming dynamic partnerships for data exchange and execute complex tasks requested by other Mist, Fog or Cloud nodes. Due to added resilience against communication instabilities the Mist computing is often discussed in connection of outdoor applications e.g. connected vehicles, intelligence surveillance etc.

Cloud computing can be also a source of data privacy concerns. The privacy of individuals is easier protected if the sensitive data does not leave the sensor, the body or the house where it originates. Once the data has left the private sphere we need sophisticated, complex and expensive technical, institutional and legal solutions to ensure a robust protection of privacy. However, if locally processed data does not leave the private sphere it is relatively easy to install mechanisms to guarantee that data is transmitted to outside agents only with explicit permission of the owner. E.g., we have used wearables with accelerometer and pulse rate measurement to differentiate between different in-door activities/behaviours with a good success rate for decision making about the physical health of the individual [2]. Still, it is an information which is not expected to leave the private sphere.

The growing concern in the IoT era is the energy consumption. Even mostly in low power mode, the 14 billion network enabled devices, which are currently in use worldwide, waste 400 TWh (terawatt-hour) of electricity per year. With 50 billion devices in 2020, the consumption is expected to increase by a factor 3.5 to over 1,400 TWh [3]. Since communication over the network requires energy, the overhead of sending data to a cloud server for processing is proportional to the amount of data and the distance. On the other hand, processing data locally requires more complex and costly nodes. This implies a complex trade-off, in particular if local nodes are powered by battery or harvested energy. While the specific shape of the trade-off curves depends on the application details, we can identify the main factors. First, the energy for local processing has to be compared with the energy for communication plus the energy for cloud processing. Second, as will be explored below under the term "attention", the possibility to tune the amount of sensed, communicated and processed data to the actual needs in a particular situation of the system, can save significant energy by avoiding unnecessary activities. Local processing has the advantage to be able to react faster to changing needs due to the omission of communication delays.

In fact, it can be argued that sophisticated local processing not only can lead to reduced energy consumption but also to improved adaptation and more robust behavior. Self-awareness holds the promise that the local node or sub-system has a comprehensive understanding of its situation and its environment which leads to better decisions on what data to collect, how to process it, what data to communicate and what decisions to take. Self-awareness implies [4, 5]

- that a device can assess the quality of the sensed data, i.e. it keeps track of precision, accuracy and completeness of the collected data;
- that the system understands its own performance, i.e. if it has performed well or badly in the recent past and over a longer time period;
- that the system understands if the environment is meeting its expectations, e.g. if it is in fact in an environment in which its operation is meaningful.

An active and quickly growing area of research [6–10] explores the possibilities, costs and implications of self-awareness in various application domains. In this chapter we will focus on the base of case studies how self-awareness can be realized in these application domains and its potential benefits.

The paper is organized as follows: The cloud, fog, and mist computing in various application areas is examined in section 2. In section 3, a self-aware data processing at fog and edge nodes is discussed. There are four related case studies - health monitoring in section 4, home patient safety monitoring and training support, described in section 5, household self- and remote control in section 6, and intelligence surveillance in section 7. The analysis of fog-level smart gateway implementation feasibility and options is discussed in section 8. The final section 9 offers concluding remarks.

2 Cloud, Fog and Mist Computing Networks

Taking a step back and analyzing why we deploy sensing systems, we realize that the reason for this is to collect data for decision making. The decisions could be made by a machine or a human, based on the type of the decision that must be made, the types and amount of data being collected and the data processing methods must be selected. This selection process is part of the data to decision paradigm, introduced by Broome[11]

Table 1. Selected attributes of Mist, Fog and Cloud computing. While Cloud computing can provide the highest (conceptual) level decisions, the limited resources of Mist computing are sufficient only for simple operational control. The Fog capabilities fall in between allowing adaptive (functional level) decision making. The figures give only an approximate order of magnitude.

	Abstraction	Distance [m]	Delay (Processing +Communication) [s]	Bandwidth [b/s]	Power [W]
Mist	Operational	10^{-2}	10^{-6}	10^{3}	10^{-2}
Fog	Functional	10^{2}	10^{-3}	10^{6}	10^{2}
Cloud	Conceptual	10^{6}	10^{0}	10^{8}	10^{5}

With the data to decision paradigm the data collection is driven by the decision making processes, from the data and information perspective it does not make any difference whether the data is being processed in the device that collects it or whether it is sent to the cloud for processing. However, many of the functional and non-functional system parameters are affected by the selection of the computing architecture when choosing between Cloud, Fog and Mist computing architectures as seen in Table 1. Some of these parameters include latency of control loops, bandwidth usage, storage requirements, security and privacy aspects, system robustness and reliability. In applications where low latency from the sensor to the data consumer (which can be an actuator) is critical Mist Computing architecture is beneficial, which is also the case when potentially large amounts of data are collected, which could be processed locally. Fog Computing architecture provides value when data from multiple sensors need to be fused and the resulting information be provided to a consumer locally, which may be the case for example when the heating or cooling requirement for a building needs to be determined based on the occupancy level of all the rooms in the building.
Cloud computing is applicable when we are either processing big amounts of data, which processing methods may be complex (e.g., data collected over long period of time, when the amounts of data are too big to be processed by edge devices or when the processing methods are too com-

plex to be executed on edge devices) or when the data sources are so distant that collecting data to a central location is feasible for example when behavioural data is collected from a city population.

Fog computing brings computation closer to the edge of the network. In Fog computing a more capable device (e.g., the gateway) bears the responsibility for data processing or IoT application execution, regardless whether the application is just simple data collection or building automation with many actuation tasks. Placing the application logic in a gateway has many advantages, such as simplicity of coordination (the centralized control paradigm used with Fog computing is very similar to conventional programming paradigms), simple management of application logic (the application logic is all concentrated in a single device) and having access to macro-level information (e.g., house or city block) in the application from all sensors. However, this approach also has drawbacks, such as increased delays in applications involving control, unnecessary high bandwidth requirements as all data must move through the gateway. The gateway is a single point of failure for applications that must be executed on the network and the operation of the entire network is dependent on the gateway.

Mist computing takes Fog computing concepts even further by pushing appropriate computation to the very edge of the network, to the sensor and actuator devices that make up the network. With Mist computing the computation is performed in the microcontrollers of the sensor or actuator nodes. The Mist computing paradigm decreases latency as devices are able to communicate directly with each other (making data directly available to the consumer) and further increases the autonomy of a solution.

When designing a solution using Mist Computing principles a monolithic architecture with dedicated device roles and interaction patterns is not feasible as it severely limits the applicability of the solution. To create a solution, which can be deployed in variable configurations and that adapts to the changes of configurations a service based architecture is optimal. By applying the principles of a service-based architecture, the application can be described as a combination of services, which are dependent on each other. Any device that has access to the network can subscribe to a service that is offered by any of the devices on the network. Hence in a temperature control application a heating unit can subscribe to temperature information from all the temperature sensors that are located in the room, which the heating unit is heating. The sensors can start providing their temperature data directly to the heating unit using the interval specified by the heating unit (the interval being dependent on the properties of the room and the power of the heating unit, and the relation of the parameters can be determined automatically at run time by the heater). No human involvement is therefore needed for setting up the application, simplifying network configuration.

Unlike a system with a fixed structure, where the functionality of the components and their interaction patterns are well controlled and predictable, in a dynamic Mist Computing scenario the interactions are not fixed as the system configuration itself is not fixed during design time. In the context of non-deterministic interactions between systems, the

temporal and spatial validity of the data that is exchanged is crucial for ensuring correctness of the outputs of algorithms using the data as an input.
Therefore, each device in the network must be aware of its location as most applications tend to be location dependent. The necessary 'location awareness' can be created at installation time (by telling the device its location), or the devices can determine their location autonomously by determining their location relative to some existing beacons, with known locations (for example a light fixture in a room may be aware of the room where it is located and all other devices in the room can determine their location based on proximity to the light fixture). Also the devices must share a common clock or there must be a way of temporal alignment of data to ensure temporal validity of data used in computations.
The services provided by end devices may also be requested by mobile devices or servers, in which case the service request reaching a specific network is routed to the device, which is able to provide the specific service. This means that in one network end devices and a gateway may be both providing services to the same server. As an example in a building automation scenario we may be interested in room occupancy information for every single room, so all the occupancy sensors in the individual rooms must report information directly to the server, while the operation times of standalone AC units may be aggregated (in the gateway) to estimate the total power usage of AC units in the building.

3 Self-Aware data processing

The umbrella term *self-awareness* encloses a number of concepts such as self-adaptation, self-organization, self-healing, self-expression, and other self-* properties. Different authors endow these terms with different, only partially overlapping meanings, but probably most will agree that self-awareness in computing devices holds the promise, that those devices exhibit more sensible behavior under novel conditions and adapt more gracefully to faults, failures and changing environments. Ultimately, a self-aware system should fully understand its own situation and detect its own misbehavior or under-performance due to

- faults, that may be caused by aging, accidents, or a physical attack,
- a malicious attack on its functions, or
- functional design errors in its hardware or software.

After deviations are detected by the self-awareness monitor, appropriate actions can be taken by are parts of the system that are typically considered outside the self-awareness monitor proper. Such actions may range from raising an alarm to an abrupt stop of all operations, or be a less extreme adaptation of behavior.
Recently, many projects have been initiated to fulfill parts of this promise. Due to the breadth and versatility of the term, its application in a wide range of different domains with various objectives and assumptions have been reported under diverse labels such as autonomic [12], nature-inspired [13], organic [14], self-organizing [15], self-adapting [16], cognitive [17] or self-aware [18] computing.

Since there is no widespread consensus on the meaning of these terms, we review briefly the properties that we consider to be part of self-awareness [4, 19, 20].

Semantic Interpretation includes an appropriate abstraction of the primary input data and a disambiguation of possible interpretations and an assessment of the reliability of the data [5].

Desirability Scale provides a uniform goodness-scale for the assessment of all observed properties.

Semantic Attribution maps properties into the desirability scale suggesting how good or bad an observation is for the system.

Attention determines which data should be collected and analyzed, given limited resources [2].

History of a Property: Awareness of a property implies awareness of its change over time.

Goals provide the context in which interpretation and semantic attribution is meaningful.

The Purpose of a smart embedded systems is to achieve all its goals.

Expectation on Environment: The system expects a specific environment and detects if the environment deviates significantly from expectations.

Expectation on Subject: Similarly, the system's own state and condition are continuously assessed to detect deviations, degradation, performance and malfunctions.

The Inspection Engine continuously monitors and assesses the situation and integrates all observations into a single, consistent world-view.

Recalling the objectives of fog computing, namely efficiency of, privacy, and reliability, it turns out, that self-awareness is apt to play a useful role. Sophisticated assessment of the system's objectives, resources, and needs, will facilitate all three goals. First, self-awareness and overall efficiency follow similar trajectories in the design space. The more complete and correct the self-assessment is the more effective will be the usage of resources with respect to given goals. A full understanding can lead to minimize the amount of computation and communication necessary close to the theoretically possible. Second, if privacy is acknowledged as an important objective by the system, the self-awareness component can track it and prevent unwarranted information leakage. And finally, maximizing local intelligence in the form of self-awareness, makes the local system more independent and resilient against disturbances in the wider system.

Hence, it can well be argued that self-awareness is a potentially significant asset in a fog computing setting, but the specific trade-offs are sensitive to the constraints and requirements of the application.

4 Case Study I: Health Monitoring

The combination of progress in sensor technology and data analytics facilitates ever more sophisticated monitoring of vital signals for medical, professional sport or leisure purposes. Inexpensive sensors for heart rate,

Table 2. Early Warning Scores derived from [22]

Score	3	2	1	0	1	2	3
Heart rate (bpm)	<40	40–51	51–60	60–100	100–110	110–129	>129
Systolic BP(mmHg)	<70	70–81	81–101	101–149	149–169	169–179	>179
Breath rate (/min)		<9		9–14	14–20	20–29	>29
SPO$_2$ (%)	<85	85–90	90–95	>95			
Body temp.(°C)	<28	28–32	32–35	35–38		38–39.5	>39.5

blood pressure, respiratory rate, temperature, blood oxygen saturation, and many other parameters can be attached to the body for inferring activities, health conditions, fitness levels and the efficiency of workouts and training sessions. As a multi-billion Euro market for vital sign monitoring with double-digit growth rates emerges [21], plenty of investment money is poured into the sector with the consequence that more versatile and less expensive sensor devices and monitoring equipment will become available in the forthcoming years. Thus, data from vital sign sensors are plentiful but must be processed quickly and efficiently.

4.1 Early Warning Score

Consider the Early Warning Score (EWS) [23, 24] system, which is a manual tool widely used in hospitals to track the condition of patients. It allows for the evaluation of risks early in order to take preemptive actions and can be defined as "a specific procedure for the early detection of any departure from normal frequencies of clinical cases or serological reactors of specific diseases by monitoring a sample of the population at risk" [25]. Based on five physiological parameters, as listed in table 2, it assigns a score between 0 and 3 to each of them, with a lower score meaning better condition. Adding up individual scores gives the EWS score between 0 and 15, which has been demonstrated to be a reasonable predictor for subsequent health deterioration and even mortality [25].
In current hospital practice EWS is a manual procedure, but recently attempts have been made to mechanize the measurement and the EWS calculation based on wearable sensors [26]. This would have the significant advantage that the procedure is not bound to the hospital any more since trained medical personnel does not have to be present for performing the measurements. Continuous monitoring of patients at home or at work becomes feasible. Anzanpour et al. have demonstrated an automated EWS system with wearable sensors, a gateway node that relays the sensor data to a server, which in turn computes the score and make the assessment. The server can be located in the hospital and medical professionals can further analyze the data and take actions as required. The advantages of this system are decreased costs, increased comfort for the patient, and increased monitoring coverage outside hospitals. The

main disadvantages are the inconvenience of carrying sensors and being wired up, and a potential loss of quality in the measurements, due to incorrect attachment, loose contacts, and faulty sensors and equipment. To address these concerns, Götzinger et al. [27] have added the capability to analyze the data reliability and consistency thus making automated EWS more robust and reliable. Indeed, a general issue with the proliferation sensors connected to the IoT in a growing number of applications and domains is the unknown quality of the collected data. Deployed sensors exhibit a wide range of accuracy and precision, hardware faults and finite battery capacity limit their lifetime, and the hardware and software of the processing and communication equipment may have their own flaws and limitations. For these reasons it is mandatory that a system like the automated EWS analyzes the quality of the collected data and keeps track of meta-information. To this end Götzinger et al. describe an agent based processing system that assesses the consistency and the plausibility of the sensor samples by a set of interacting agents that specialize on several tasks: abstraction, history tracking, confidence derivation, and binding [27]. They demonstrate that several typical failure conditions in the data collection and processing chain can be correctly identified to improve the overall robustness of the EWS system.

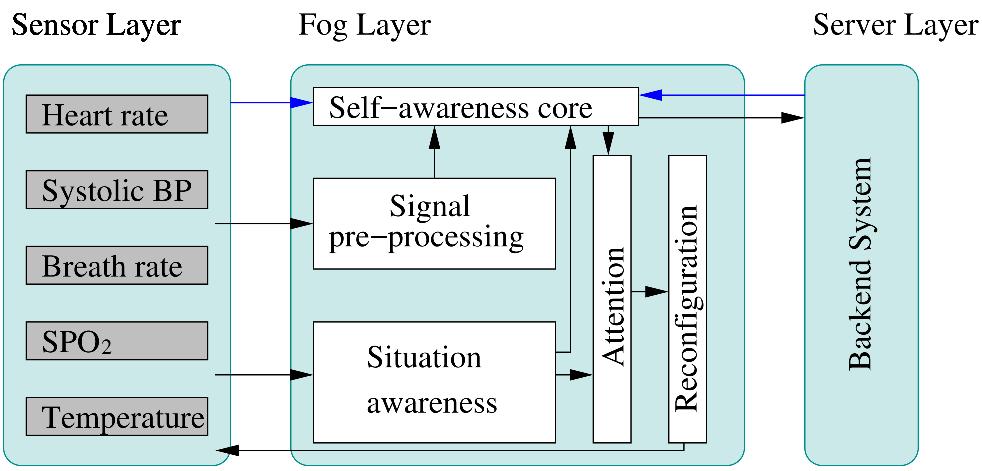

Fig. 1. Architecture of the automated EWS system [22].

Anzanpur et al. take this approach a step further and propose an architecture that combines self-awareness, situation awareness, an attention mechanism, and an adaptive resource allocation scheme, illustrated in figure 1. Based on the sensory input and the system's expectation on itself and its environment a model of its own performance and relevant aspects of the environment are built up. For a correct computation of the EWS the sensory data is pre-processed with traditional signal processing algorithms to suppress noise and extract relevant features. The

pre-processing stage offers abstractions relevant to the medical application of the raw data to the upper layers in the processing chain. The self-awareness and situation awareness blocks in figure 1 adjust the primary input data in two ways. First, an ambiguity and consistency analysis identifies potential errors in the input data. Second, the activity of the patient is estimated, since the data interpretation is dependent on the activity pattern. The system distinguished five different activity modes: sleeping, resting, walking, jogging, and running. Hence, taking both measurement incorrectness and patient activity modes into account, the most like interpretation of the data is derived and the corresponding EWS value is computed.
Furthermore, the built-in attention mechanism allows for efficient resource usage without compromising the core objective which is a correct assessment of the patient's health. The system considers four levels of severity of the patient's condition (normal, low, medium, high) five states for the activity (sleeping, resting, walking, jogging, running), and four different situations for the environment (indoor night, indoor day, outdoor night, outdoor day). Depending on the position in this three dimensional space, priorities are assigned to the activities of the system and resources are allocated accordingly. The attention block in figure 1 computes the priority vector which is then used by the reconfiguration configure the sensors and allocate resources properly. Its main objectives are, first, to maximize the quality of assessment, and second, to minimize power consumption. Possible parameters for sensor configuration are sampling rate, sample precision, bias and calibration values, activity modes like sleeping and active.
The backend system in figure 1 might include a cloud server and medical professionals. It can do further detailed analysis, initiate or fine-tune medication, or request the patient to visit the hospital. For us most interesting is the possibility to provide feedback information to the self-awareness module in the fog layer. This feedback information can be the basis for learning, adaption and improved performance of the system.

4.2 Discussion

The automated EWS system as described in [26, 27, 22] illustrates the benefits trade-offs of a sensitive and informed decision making in the fog layer. A self-awareness module as described above can increase the robustness of the data analysis and the quality of the assessment of the system's own situation, the monitored subjects state and the environmental conditions. This in turn is a solid basis for correct decisions regarding the collection of further information, triggering emergency actions and a prudent usage of resources. Obviously, all the described functions can be realized in the server layer on a main server or a cloud computing infrastructure. However, several aspects have to be considered in this trade-off.

Energy efficiency By no means is it obvious which alternative is more energy efficient. Gathering a lot of processing and intelligence locally in the fog layer can reduce communication costs between sensors and the cloud server by orders of magnitude. However, computation may be more efficient in a high performance server infrastructure because of an optimized processor architecture, large caches, and deployment of the latest processor family. Also, if the fog layer runs on battery a fraction of the energy is lost in non-ideal batteries. On the other hand, local processing allows for customization by adjusting to the actual precision needed, by avoiding unnecessary generality in the architecture, and by eliminating unneeded computation altogether. Hence, we face a complex trade-off and the most energy efficient solution may be somewhere between the extremes of all-local and all-central computation and depend heavily on the specifics of the application.

Latency Likewise, the latency and response times depend on a variety of factors. To begin with, not every application is particularly latency sensitive. If it is, it makes a difference if average or worst-case latency is more important. On one hand, fog-level computation avoids the delays of communication between the local node and the server, but on the other hand, the server may be significantly faster in doing the required computation. Hence, again, the optimal solution with respect to latency is application dependent and may be located between the extreme points of the design space.

Customization of the hardware architecture and software can lead to significant gains in terms of optimality and efficiency. The downside is that customization requires effort in design, validation and maintenance. Hence, the optimal point cannot be located in general terms, depends on the application and will in most cases end up somewhere between the extremes.

Location of control If most of the processing is performed in the fog layer, it has two implications which concern efficiency, reliability and privacy: First, the locally generated data never leaves the local environment, such as the person's private home. Usually, only abstracted data is sent to the main server and only in specific cases a full record is requested, for instance when the case is unclear and a physician wishes to examine the details of the situation. Second, local configuration options, such as which data to sample and to store, is decided locally and not by the remote server. Locality of data and decision is an efficiency issue for the reasons discussed above under the terms energy efficiency and latency. It is a reliability issue, because if even mundane and simple operations like counting the heart beat is dependent on a flawless internet connection and server infrastructure, which makes the system vulnerable to many kinds of disturbances. Finally, it is a privacy issue because only the fog-layer alternative provides the option that the user can control what data and decision leaves her private sphere. When data and configuration access rights are sent and stored at a server, it requires more elaborate and stringent policies to protect privacy concerns.

5 Case Study II: Patient safety monitoring and training support

The cost of hospitalization and elderly social care is increasing worldwide. Telecare is believed to compensate the reduced traditional clinical interactions and home nursing. IoT technologies certainly have a role in traditional telemedicine of chronic disease management as well as guaranteeing patients' safety at home, enabling new services like technology assisted rehabilitation. The market research company Gartner expects that healthcare related technologies and services will have 16 percent of IoT market business value in 2017.

Distributed and mobile sensing devices are well suitable for improving safety of elderly and special needs patients' at home. According to Centers for Disease Control and Prevention about 800000 patients are hospitalized every year in USA due to a fall down [28]. Ibid., 25% of older adults will fall every year in USA. In addition, it has been shown that daily activities and cognitive and physical health might be in good correlation [29][30].

Certain modern home telecare systems already provide of fall down detection functionality based on wearable motion sensors, camera systems, floor sensors [31]. Technologically it is also possible today to predict the frailty of elderly people based on activity of daily living (ADL) pattern analysis [32]. However, compared to conventional telecare solutions that just periodically transmit vital signs measurement data, the real-time and dependability requirements for such safety critical telecare solutions are certainly higher. By conventional home telecare systems the measurements are usually conducted twice a day and even by ECG signal measurements the recording package does not exceed 50 kB. Therefore traditional central server or cloud based data store for personal health records (PHR) has been sufficient so far. The technological and also privacy requirements for novel home telecare solutions employing certain motion capture or continuous ADL monitoring capabilities are significantly stronger which leads to use of alternative - distributed data processing architectures. It can be estimated that the raw data stream of an inertial measurement unit (IMU) capable for sufficiently accurate human activity or free fall tracking is at least 1 kbps. Similar average data stream can be expected from low frame rate safety observation cameras. It is simple to understand benefits of distributed fog-like data processing for such real-time patient safety monitoring solutions over the centralized ones. The remote server load and communication channel throughput can be significantly reduced through the local data aggregation and decision making. Also, from the clinical point of view, only the aggregated metainformation i.e. number of active hours, mean activity level, sleep quality and presence of special events like fall downs have significant long term value worth for preservation in PHR. The rest of raw data has a low clinical value and rises the privacy concerns if the data is transmitted outside of the private areas.

Nowadays the smartphones are frequently used as telemedicine gateways and wireless communication is widely used. For the majority of sensor signals i.e. temperature, conductance, movement and position, illumination, the wireless transmission consumes 100-1000 times more energy than processing that is also motivating the local data aggregation. Distributed fog computing also increases the dependability of wirelessly networked systems. Today Bluetooth Low Energy, ANT+ and different IEEE802.15.4 standard compliant radios are mainly used for the personal area networking. Due to the throughput limits the real-time data streaming may seriously affect the reliability of the communication. Fog computing reduces such risks as well because because of relaxed requirements for the communication channel throughput. Even more, if the real-time critical data processing is done locally, redundunt communication channels i.e. over wireless MESH network can be effectively used. A typical distributed ADL safety monitoring system is shown in Figure 2.

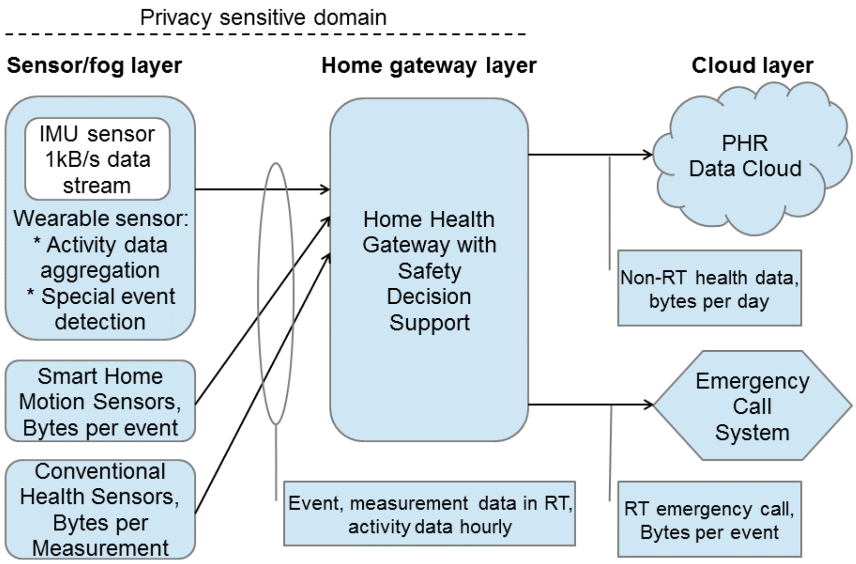

Fig. 2. ADL safety monitoring and estimated data rates of various I/O channels.

Theoretically it would be possible to acquire required user activity, location and falling information from one wearable IMU device and process the data within the same sensor device. Modern IMUs usually have built-in free fall event detection. Combined use of linear accelerometer, gyroscope and magnetometer data should be sufficient for dead reckoning based movement monitoring. In practice sensor fusion with alternative smart home sensors still has to be used because of internal errors of inertial motion sensors. In real life human fall down cannot be sufficiently reliably detected with inertial sensors only and due to nonlinearities of IMUs the dead reckoning movement tracking is reliable just for some meters. Due to the unpredictable network delays the described sensor

data fusion cannot be made at a remote location. Therefore the sensor data fusion, fog-based aggregation and possible reasoning at the gateway device is the most appropriate solution for intelligent home telecare systems supporting ADL analysis and hazard detection.
Reablement through the support to physical activities extends the independent living time and therefore reduces the needs for expensive traditional social care. For example, it has been reported cumulative cost savings of 30% in 2-5 years through the trainings [33]. However, reablement process itself that includes human assisted trainings and validation of new skills is costly and time consuming. It is expected that wearable sensors and other IoT devices will enable teletrainings and safety validation of home-basedexercising.

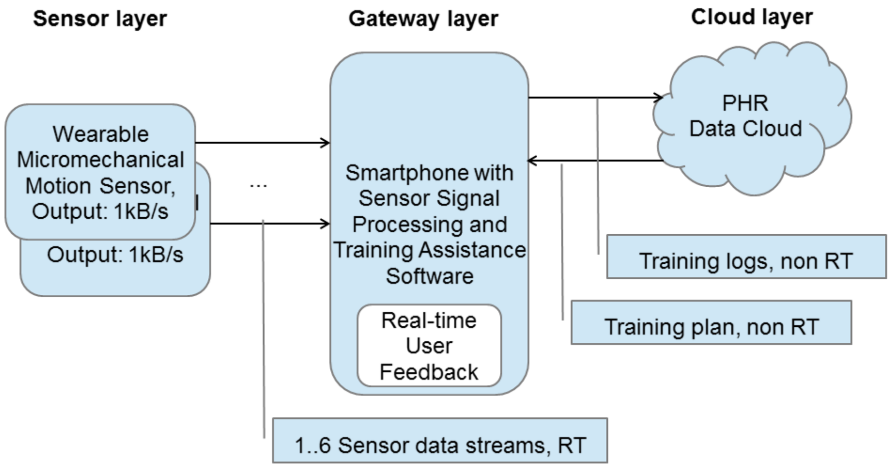

Fig. 3. Home-based training assistance.

Remote validation of physical training exercises through the home telecare systems may be possible the near further. Such technological solutions would significantly reduce the needs for physiotherapists and clinical visits. For example, specific home exercising is required during the stroke recovery [34] and after joint replacement surgeries [35]. In both cases rather simple exercises has to be performed to avoid irreversible processes of joint stiffening. For the training efficacy certain exercising speed and amplitude have to be preserved which was not possible earlier. Today the training process may be well monitored with wearable IoT devices giving real time feedback to the user if the exercising is accurate or not. It would be logical to connect such training assistance unit with the above described home telecare system, which will effectively enable the targeted machine-assisted exercising. Essentially the system should transmit the quantity and quality of performed training exercises to the PHR cloud server where the clinicians and physiotherapists can access the data and make further treatment decisions as seen in Figure 3. As in a previous ADL monitoring example, it is feasible to process the sensor data locally in the fog to minimize the load of communication channels and save resources of PHR repository server. In this particular case the decision about exercising correctness has to be essentially made on-site. Local decision support is required to provide exercising feedback in real-time, without noticable latency, and to meet safety critical dependability needs of the full data path. The most practical is to implement training assessment processes in the gateway device which usually has sufficient amount of the computing power and can access to the context information regarding the environment and user. Fog-based local decision support also preserves the user privacy because minimizes the personal information amount to be transmitted to the remote locations.

6 Case Study III: Smart House

In the chapter we analyze the experience and lessons learned from the multi-year use of a smart house solution. The focus is on smart house energy management (HEM).

Main principles used in implementation of this smart house solution are following:

- Computations are made up close to the sensors and actuators, both for security as well as the latency reasons. Home gateway (a fog computing node) is the place where they are held. The Cloud computing is used for analyzing big amount of data and for decision models developing and testing.
- Self-Awareness is widely used. System adapts to the specific sensed situation. The system is very cautious with respect to non-authorized objects. Process of the authorization the object is two-step - at first, the system administrator has to introduce the device to the system, and only after that the device will be added to the system. The system continuously scans the devices accepted and in case of mismatch refuses to interact with the device.

Abowd and Day [36] introduced the primary context types:

- Location – location data from the GPS sensor,
- Identity – Identify object based on the RFID tag,
- Time – read time from clock, also daytime,
- Activity – what activities are in progress?

There is also a secondary context type, which is derivative information that we can use based on the primary context. For example, using Identity, we can obtain considerable information about a person in social networks and the internet.

In our case, the context information is the place where we can begin our analysis. Our system has ability to react to the new situations and learn from results. For our approach, the adaptive and self-organizing properties are exceptionally valuable. Instead of attempting to find the most optimal solution based on the available information, we attempt to use self-organization methods. For example, it is known that house energy planning is a highly complex and difficult problem. The situation

can change notably quickly, and our perfect plan fails. Instead of careful planning, we attempt to use adaptive techniques. For example, we can obtain information about energy needs from the sensors and quickly use stored energy (Tesla Powerwall).
Before we go to real examples lets look at the concept of the smart solution.
The smart-solution concept is notably important in the context of this example. Smart refers to quick intelligence. People are considered smart when they show up a rapid ability to respond intelligently to different situations. As we observe, smartness is strongly connected with the concept of intelligence. It is a long debate regarding whether we can exhibit the intelligence to computers or software. Todays computers can do many things that require intelligence, such as driving a car off-road or on city streets.
The term intelligent systems is used to describe the necessary level of capabilities to achieve the system goals. Intelligence has been scientifically categorized as a biological stimuli response mechanism. In our case, we obtain the stimulus from the environment using different sensors and make a response using the knowledge that we have and the actuators that are connected to the system. During her lifecycle, the system learns from experience. The learning ability is precisely what makes the system intelligent. Computer power and the amount of information and sensors make a system smart.
Smart solutions are composed of smart objects [37]: *"One definition of smart objects is a purely technical definition – a smart object is an item equipped with a form of sensor or actuator, a tiny microprocessor, a communication device, and a power source. The sensor or actuator gives the smart object the ability to interact with the physical world. The microprocessor enables the smart object to transform the data captured from the sensors, albeit at a limited speed and at limited complexity. The communication device enables the smart object to communicate its sensor readings to the outside world and receive input from other smart objects. The power source provides the electrical energy for the smart object to do its work."* These objects can learn and adapt to different real-world situations, and different machine learning algorithms are used [38].

6.1 Case study object characterization

This is a six-member family living house, where electricity use is high as seen in Figure 4. The system has two heat pumps, gasoline power generator, and three oil radiators. The largest consumers of electricity are water heater, washing machine and stove. The house usage is irregular because the residents are working adults, and some days they do not use the house. All this makes it difficult to optimize energy use, but gives a great economic effect.
Traditional approach to make house energy systems is to define system requirements, designing solution and implement solution by professionals team. All this is costly and needs time to get results. Typically, the end user will not be able to actively influence the system behavior and functionality. In our approach, we choose another path.

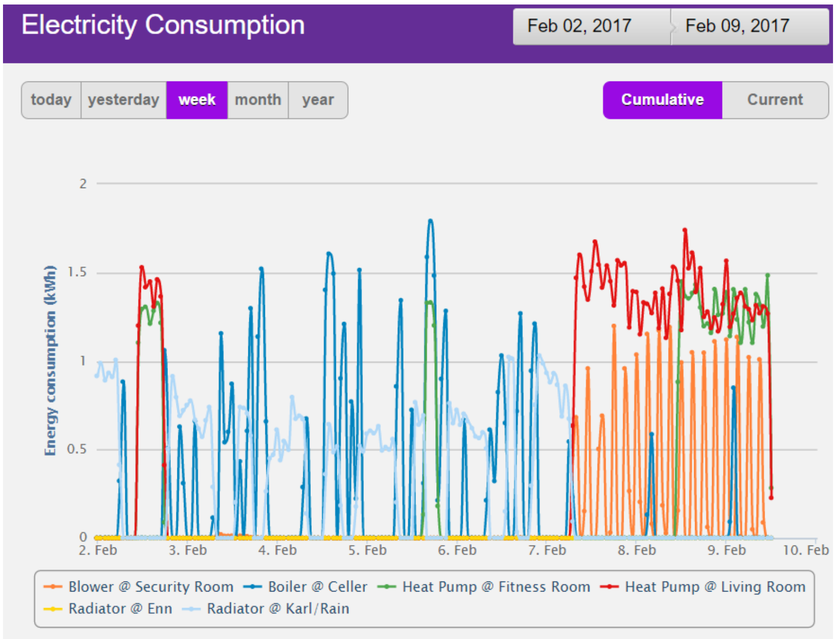

Fig. 4. Typical weekly electricity consumption in house.

1. We enable intelligent decisions by liberating device data across operations.
2. Decisions are defined by user using rules.
3. System collects decision data with context information, analysis information and makes recommendations.

One example that can be used to optimize the use of energy, is the heating device duty time. The aim is that upon returning home, the family has a house comfortably warm, while the heaters are not turned on too early. The house settings user interface[4] is presented in Figure 5. The interface has both informational role (room temperature, light level, humidity), as well as the characteristics of actuator triggering role. For example, if the user clicks a camera icon, he/she gets the camera video stream or in the case of clicking a plug icon the light or heat pump will be toggled.
Users of the smart house are very satisfied with the solution due to number of reasons:

- Security. The house has a security system to detect and alarm in case of smoke/fire. There are day/night surveillance cameras to monitor the household in habitats absence.
- Convenience. The house itself (heating, airing) is continuously tuned according to user needs.
- Money savings. Controlled heat pumps and optimized usage of the water heater reduce costs.

[4] Telia Eesti AS https://www.telia.ee Smart Home solution. The service marketing has been discontinued since 2017

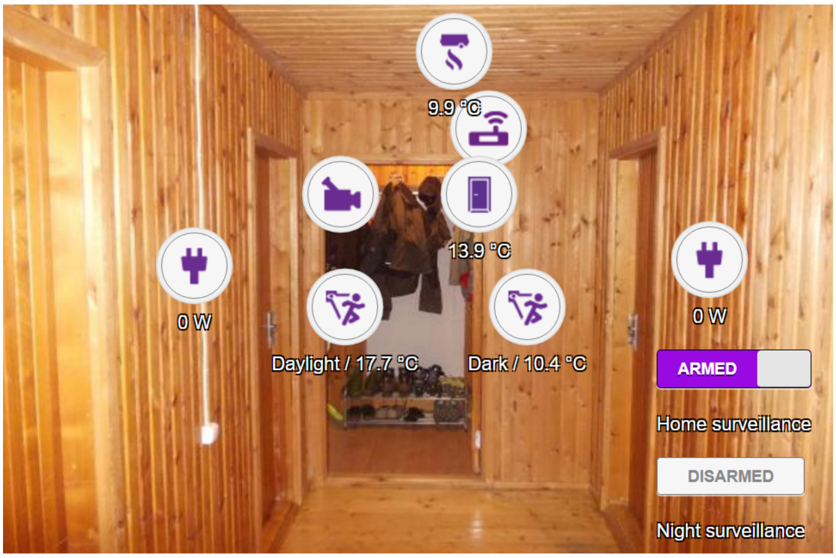

Fig. 5. Family living house settings user interface.

We plan to complement the solution to the solar panels, energy storage and energy purchase from the market. It is also planned to set up a small-scale local energy grid.
In conclusion, it can be said that the principles set in this section, turned out to be fruitful. Decisions should be taken as close as possible to the equipment following the fog computing paradigm and using as much as possible the context information. Only in case of large volumes of data the advantages of cloud computing should be exploited.

7 Case Study IV: Intelligence Surveillance, Reconnaissance - military sensing systems

As stated in the beginning of the chapter military sensing systems are also undergoing an architectural change, driven by the enhanced capabilities of the system components. While Network Enabled Capabilities (NEC) was a compelling vision in the very beginning of the century, we can say in 2016 that the technology components and architectures are catching up with that vision. This section describes an Intelligence, Surveillance and Reconnaissance (ISR) solution concept, which builds upon the concepts of Mist and Fog Computing and that is very close to the NEC vision introduced more than a decade ago.
In the European Defense Agency's IN4STARS project the Research Laboratory for Proactive Technologies developed an ISR solution prototype, which relied on Mist and Fog computing methods. The task of the solution was to provide enhanced situation awareness for units in the field and to remote intelligence operatives. The sensor system deployed in the

field processes collected data locally at the sensor level using Mist and Fog Computing principles and delivers only situational data and information that has been requested by the users to the users, following the Data to Decision [11] paradigm.

Unlike the classical system approaches that assume a central coordinating agent, the sensor system architecture applied in the project builds upon a mixed Fog and Mist computing approach, where the individual Mist Computing nodes are autonomous. When a request for information is made to a deployed ISR system, any node that is capable to provide the requested information with an acceptable cost will respond to it. The specific sensor modalities needed for providing the requested information (e.g. detection and identification of tracked vehicles) need not be co-located with the system providing the information, instead the information may be fused from several sources, including both ground based as well as airborne sources. To enable this kind of operation the nodes must maintain a certain level of self awareness as well as awareness of the system itself, in order to find the required sensor sources for generating the information requested by the information consumer. In order to achieve and maintain the required self and group awareness the individual systems must be able to communicate directly and to request services from other systems. The conceptual system configuration is depicted on the Figure below.

Applying the D2D approach in a Fog Computing paradigm means that the requests for situational information made by the information consumer can be directed to the sensor assets in the field, closest to the area of interest. The routing of information to the specific information provider may be done using many alternative methods, e.g., geo-routing, using a central service directory or some other service discovery mechanism. Based on the information requests, the algorithms are primed in the computing device providing the information service (e.g., sensor or fusion node). Service requests are made to the data sources (sensor nodes) from which data is needed for computing the requested situational information. Once the information has been computed it is provided to the consumer.

The sensor system was built as a wireless sensor network with a dynamic network structure and functionality and ad hoc communication paths. Depending on the information request received by the system the appropriate Mist and Fog Computing algorithms are triggered locally to process the collected sensor data and to deliver the requested information to the user. Every sensor that was part of the solution was equipped with a local computation unit and a wireless communication interface, making it an independent node in a Mist Computing solution. The multimodal senors (seismic, infra-red, acoustic, visual, magnetic, etc.) employed in the solution made use of in-sensor signal processing, including novel detection and classification algorithms based on in-depth analysis of sensor data. The solution comprised several sensors, which were assembled into a system in an ad-hoc manner, enabling real-time configuration and behavior adaptation.

In order to assess the locations and types of the detected objects with a higher precision and to service the situational information needs of

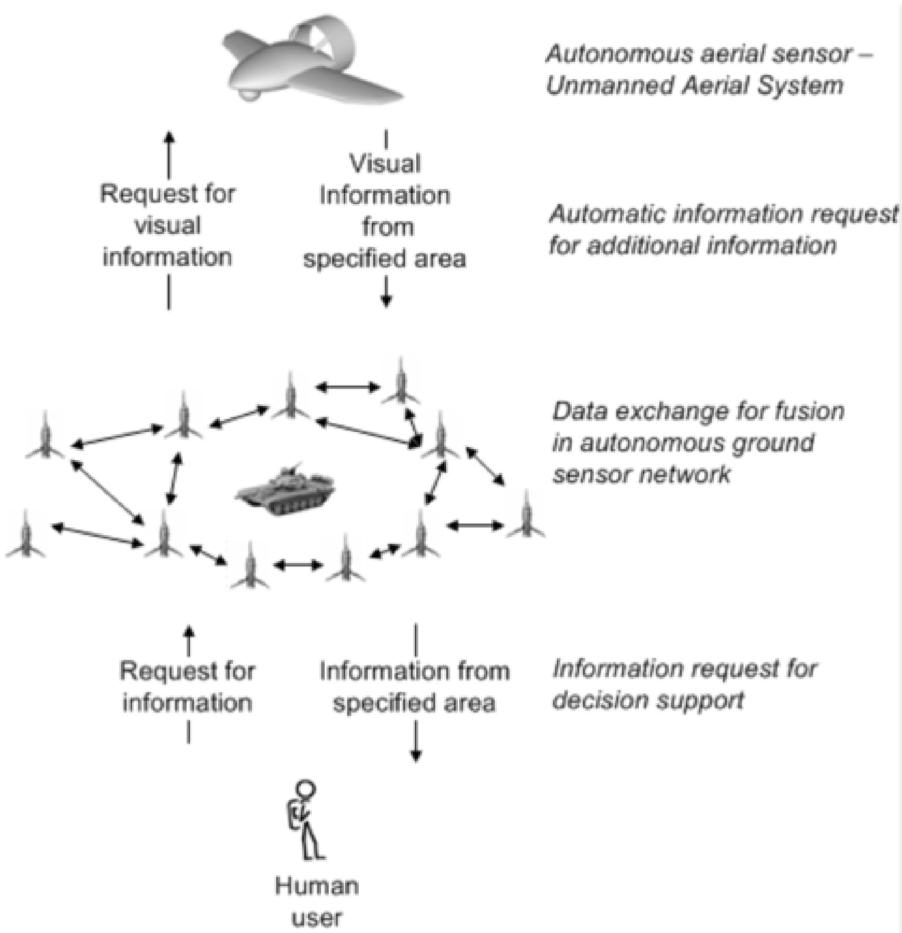

Fig. 6. Architecture of the automated EWS system [22].

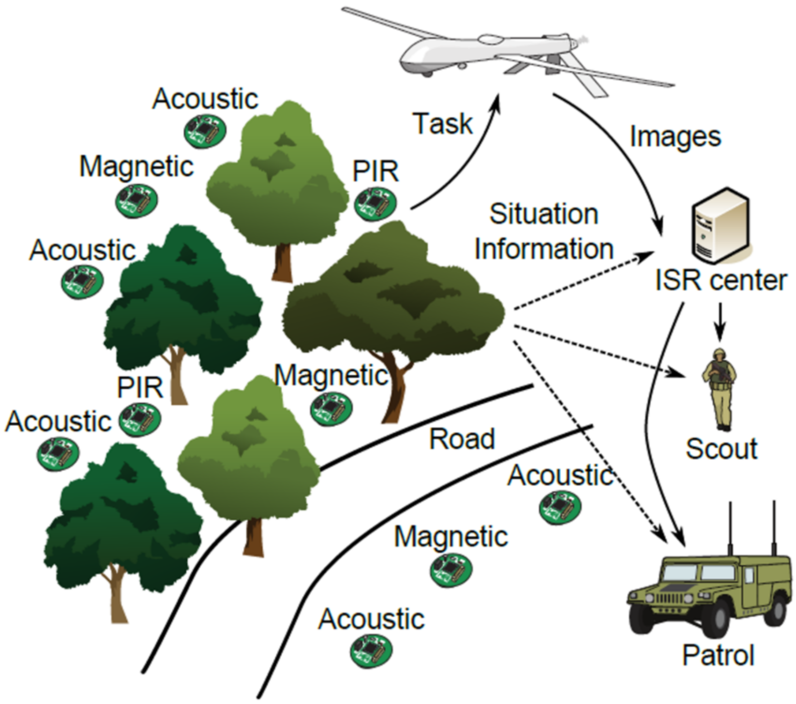

Fig. 7. Architecture of the automated EWS system [22].

users a Fog Computing layer was added, which performed in-network distributed data aggregation and fusion, which relied on a service-oriented middleware based on a subscription model. At the Fog layer data fusion methods were applied, with on-demand data to sensors, collecting data needed for a given fusion operation based on the information requests that had been made by the user. This logical system structure enabled hierarchical buildup of situation information by distributed fusion and aggregation. In order to achieve correctness of data time selective communication was used to provide temporally aligned data for fusion and aggregation algorithms at the Fog layer.

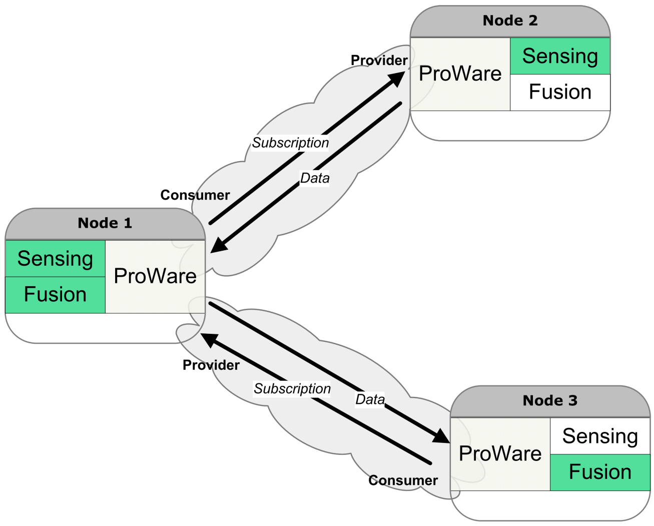

Fig. 8. Architecture of the automated EWS system [22].

The distributed processing implemented using Fog and Mist Computing principles was enabled by the Proactive Middleware, which has been developed at the Research Laboratory for Proactive Technologies at Tallinn University of Technology. ProWare offers the services of data provider discovery, on-line data validation and service contract agreements between data providers and consumers. Such an architecture facilitates predictable operation also in a changing system configuration. With these features ProWare enables dynamic creation of data and information exchange relationships in a distributed network using a subscriber model. ProWare solves one of the major challenges in a distributed computing scenario, which is ensuring the temporal and spatial validity of data - making sure that the data used in computations originates from the right location (i.e., from the correct sensor node) and is temporally coherent with other data used in the computation. The latter is very critical in sensor data fusion applications, but difficult to achieve when the fusion is performed in a distributed manner using Mist and Fog principles.

The sensor systems that made up the ISR prototype can be categorized to ground sensors and airborne sensors, below both types are described and the operation of the systems discussed. The object localization solution based on acoustic arrays utilizes autonomous acoustic arrays working together for localizing detected objects. The same arrays can be also used for acoustic classification using any of the available classification methods as we have also presented in our previous work [39].

8 Requirements and architecture for a smart gateway based on hierarchical temporal memory

Common in all case studies above is the need for an intelligent processing unit capable of learning, model building, behavior adaptation, and anomaly detection. The backpropagation learning algorithm in traditional NNs is slow [40] and requires intensive use of floating point arithmetic (e.g. to calculate sigmoid function), which makes it ill suited for applications where 'human-like' quick reaction and continues learning ability is expected.

Among contemporary continuous learning algorithms the Hierarcical Temporal Memory (HTM) of Numenta, inspired by the architecture of the neocortex, has proved to be successful in comparison with many other detector algorithms [41]. Moreover, the Numenta approach is attractive due to a smaller computational load where instead of complex floating point calculations simple fixed point and integer arithmetics is used. This property makes HTM attractive to be tested in application areas where memory, processing performance and energy are limited.

In this section we discuss an approach for the design of HTM based microcontroller and SoC computing platforms for fog computing gateways, to meet the required processing capabilities, memory, and energy consumption.

As illustrated in Figure 9 we consider gateways responsible for the following functions and actions:

- Edge input devices control and power management;
- Input data fusion and time-stamping;
- Data abstraction and conditioning;
- Encrypted data exchange with cloud service;
- Data encoding to Sparse Data Representation format for HTM processing (spatial and temporal pooling of data);
- Continuous/dynamic model building, prediction and anomaly detection;
- The first level reasoning and decision making about the situation;
- User feedback and interaction (Human-Computer Interface)
- Introspection: The service quality over time and the history of actions.

Microcontroller/microcomputer platforms might be capable for aggregating sensory data and run the HTM (Hierarchical Temporal Memory) based data processing (prediction and anomaly detection). As an example of mobile platforms, even the ARM microprocessor based smartphones could be viable solutions due to great connectivity and abundance of memory. Still, the on-line massive computations required by

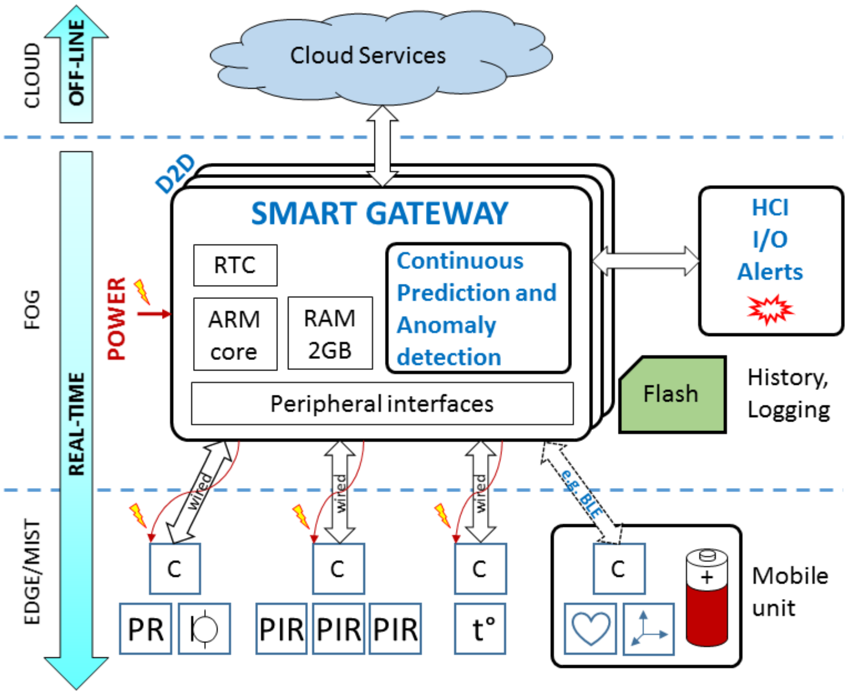

Fig. 9. Smart Gateway according to the Fog computing paradigm. C - controller, PIR - Passive Infrared Sensor, PR - Photo-resistor, D2D - Device-to-Device communication, HCI - Human-Computer Interface.

HTM might be largely impeded due to multiple regular applications of the smartphones. Their energy reserve can't guarantee 24/7 readiness and in addition, continuously on-line devices pose a risk to confidentiality and privacy [42]. Due to that stationary, single-board-computers like Raspberry Pi[43], Pine 64 [44] or Odroid C2 [45] might be preferred.
Currently, the HTM tool NuPIC can be compiled to run on desktop or laptop platforms, there are limited attempts to get it working on single-board computers like Raspberry Pi 3 or on an Android platforms, which are actually the most attractive gateway modules in Fog computing due to their low price, high availability and good community support. As a result, NuPIC is available for RasPi 3 platform only since June 2016[46].
Despite mentioned above positive properties the pure software implementation of HTM is resource demanding. Although the basic operations are simple the need for memory and performance in real-time situations is high (GigaBytes of memory, multicore processors etc). Due to that nature the HTM is not applicable at far edges of the network but it is rather a gateway level of functionality to manipulate higher levels of real-time data which is already pre-processed, filtered, averaged, and fused from raw data at edge sensor nodes. Still, there is no good comparative study of energetic aspects of different NN and HTM algorithms.
HTM functionality equipped gateway might be used to learn and assess the patient safe behavior at home. Still, the behavior of humans is in great extent defined by environmental and social situations outside of household, depending of the media influence, environmental conditions, political situation, emerging trends of society etc. All this in various aggregations is influencing the daily human behavior. Therefore, the reasoning abilities of the local (in-house) processing gateways or nodes remain always limited due to lack of holistic information i.e. the lack of the "big picture". The higher level predictions has to be done in cloud environment which does possess necessary information, reflecting all possible ramifications of the general situation back to the gateway nodes.
The computing capacity problem will be relaxed as the processing capabilities of microcontrollers are improving following the Moore's law. The same is valid in case of various systems on chips (SoC). E.g. the high-end programmable logic SoC-s from Xilinx Zync-7000 family have internal block memory capacity till 26.5 Mbits (dual-port, programmable, built-in optional error correction) also they carry Dual-Core ARM Cortex-A9 microcomputer IP [47].
In SW implementation of HTM a neural connection i.e. synapse is a data record consisting of an address of source neuron along with additional dynamic information e.g. permanence and activation history. Here, the permanence represents the stages of growth of the synapse [48]. All this information has to be present in working memory to guarantee efficient processing.
As an example, the Numenta NuPic HTM model tool NuPIC [48] processing network consists typically of 2^{16} neurons, which is matching well to number of neurons in mammalian cortical column at layer 3 [49]. Thus, an average HTM module (approximate equivalent of cortical column layer 3) consists of 2048 (mini)columns, 32 neurons per column i.e. $2^{16}\ neurons \times 20\ dendrites\ per\ neuron \times 40\ synapses\ per\ dendrite$. All

together, 52 428 800 words are required to record all possible connections (synapses) in a network. When using regular 32-bit word length, where 16 bits are used as source neuron address of the synapse and 16 remaining bits as a permanence and the activity history value of the synapse then the minimum amount of allocated memory is 209 715 200 bytes.

Modeling more complex neurons with hundred of dendrites each with hundreds of synapses the memory requirement can easily exceed a gigabyte barrier. It is still far from the capacity of top neurons of a human cortex - a single pyramidal cell can have approximately 12,000 dendrites and receive around 30,000 excitatory and 1700 inhibitory inputs [50].

In addition there are values to represent column activation, state of every neuron, threshold values for dendritic segments and columns etc. Finally, there are parameters helping to control and distribute loads over all minicolumns and neurons in the network. Processing load is not as serious problem because due to sparsity principle only 2% of neurons will become active after the spatial pooling stage, predicting the next state (input).

A block memory of Xilinx ZYNC SoC with size 8KB allows to record and keep connectivity information about at least a single artificial neuron with its many hundreds of connections (assuming each word to represent source address of synaptic connection and permanence value). The top FPGAs of Xilinx Zynq-7000 series contain many hundreds of block memories (755 in "high"-class XC7Z100, 140 in "consumer"-class XC7Z020) which allows HW implementation of regular HTM. Still, the connection resources on FPGA might become an obstacle although inter-neuron connections can be implemented using Time Division Multiple Access (TDMA) manner using serial lines only. Integrated multi-core ARM microprocessors are capable of processing even larger amounts of neural processing data in external memory, the drawback is remarkably higher energetic cost.

Still, the NN processing in regular HW/SW technology is power-hungry and not suitable for applications where available energy and memory space is limited. Designer of the power limited system has either decrease the available "smartness" of the nodes or insert learning capable nodes only to locations with low-speed real-time requirements (allowing majority of time to exploit the sleeping mode), leaving power hungry tasks to higher level gateways or servers with abundance of power.

As an example, a self-driving car has to process huge amount of high-resolution mapping, visual, radar and sensory real-time data within tiny fraction of second (faster than a human driver) to guarantee safe driving in unpredictable traffic conditions. In contrast, the processes inside artificial or natural living environments are safe when sampling and decision making intervals have period in seconds or even minutes. In such a circumstances the usage of sleeping mode allows drastically reduce the overall power consumption (peak power demand has to be satisfied, of course). Using analogy in living beings taxonomy then self-driving car processing speed could be comparable with processing speed of cheetah visual cortex, processes in living environment are advancing in turtle-speed. In insect world taxonomy the extremes might be the fly and caterpillar.

The typical 8- and 16-bit microcontrollers used at the edge of the system (in sensory nodes) are capable only for sensory signal filtering, rudimentary fixed algorithm processing and communication with higher level nodes due to very limited memory, processing and power resources. Still, quite complex local regulatory functions are possible at edge nodes, e.g. various PID-controllers fall into this category.
E.g., the highly popular 32-bit ARM Cortex-M (microcontroller processor) family devices are well suitable for home automation tasks due to integrated peripherals and power efficiency. The only problem is a limited integrated memory (up to 256KB SRAM in TM4C129x MCU series [51]), which restricts their usability in neural or cortical learning type of algorithm based applications where the number of tightly interconnected neurons alone might be measured in thousands.
Often more than a single input signal has to be followed to discover an abnormal behavior of the subject. E.g. in case of rehabilitation care the inertial measurement units (IMU) are used to assess body part movement. The movement might be wrong in any of six degrees of freedom, also along the time dimension (too slow or too fast), which is difficult to represent as an algorithm. Instead of that six separate HTMs, implementing six cortical columns, might be able to follow all signals concurrently and the decision making can be based on aggregated anomaly score of those HTMs.
The HTM is not ready to exhibit full capabilities of the mammalian cortical column. Still, it has an useful set of properties to be exploited at fog level gateways. The HTM enriched gateways can be responsible for input abstraction, classification, anomaly detection and communication with cloud level processing nodes, describing the observable situation in high level abstract terms like 'normality', 'anomaly', 'alertness', ' attention', health', etc.

9 Conclusion

In this chapter we have confirmed the observation of many authors that hosting signal analysis and some intelligence in the sensor nodes (edge/mist computing) and the local gateway (fog computing) is beneficial in a variety of applications with respect to performance, reliability and privacy. The emerging concepts of self-awareness promise to bring a new level of sensible and adaptive behavior to the local sensor and gateways. In all four case studies these potential benefits of local intelligence are apparent even if they have not been fully realized. However, although the qualitative arguments for distributing computation among the hierarchy levels are compelling, serious challenges remain:

- The involved trade-offs are poorly understood in quantitative terms. Moving a piece of computation from the cloud to the smart gateway or to the sensor node involves a significant change in the communication needs but also in the computation efficiency, because computing platforms are radically different in the different locations. A holistic trade-off analysis will depend on the details of the application, the involved platforms and the protocols. It has rarely been done for specific cases and is altogether missing in a generally applicable way.

- Although we have many examples of sensor node and gateway architectures, there is no common view on what resources are required and how the architecture should be organized. Some platforms have become fairly popular, but it seems that the field is moving quickly and requirements are shifting. Thus, convergence on one or two winning platforms is not imminent.
- No method with supporting tools have been proposed that can guide a an application engineer through the design of application while exploring the trade-offs due to the choice of platforms, functionality, and the distribution of computation across the hierarchy of mist, fog and cloud computers.

In this chapter we have illustrated the benefits of intelligence and self-awareness in mist and fog computing and we have sketched a possible architecture for an gateway that meets the requirements as we understand them today. However, this also highlights that there are significant challenges and work to be done in this still young but very dynamic emerging field.

References

1. A. V. Dastjerdi and R. Buyya, "Fog computing: Helping the internet of things realize its potential," *Computer*, vol. 49, pp. 112–116, August 2016.
2. J. S. Preden, K. Tammemäe, A. Jantsch, M. Leier, A. Riid, and E. Calis, "The benefits of self-awareness and attention in fog and mist computing," *Computer*, pp. 37–45, July 2015.
3. "More Data, Less Energy: Making Network Standby More Efficient in Billions of Connected Devices," https://www.iea.org/publications/freepublications/publication/more-data-less-energy.html, August 2014.
4. A. Jantsch and K. Tammemäe, "A framework of awareness for artificial subjects," in *Proceedings of the 2014 International Conference on Hardware/Software Codesign and System Synthesis*, ser. CODES '14. New York, NY, USA: ACM, 2014, pp. 20:1–20:3. [Online]. Available: http://jantsch.se/AxelJantsch/papers/2014/AxelJantsch-CODES.pdf
5. N. TaheriNejad, A. Jantsch, and D. Pollreisz, "Comprehensive observation and its role in self-awareness - an emotion recognition system example," in *Proceedings of the Federated Conference on Computer Science and Information Systems*, Gdansk, Poland, September 2016. [Online]. Available: http://jantsch.se/AxelJantsch/papers/2016/NimaTaherinejad-FedCSIS-ComprehensiveObservations.pdf
6. "IEEE International Conference on Self-Adaptive and Self-Organizing Systems," 2007-2016.
7. J. Pitt, Ed., *The Computer after Me : Awareness and Self-Awareness in Autonomic Systems.* Imperial College Press, 2014.
8. "SelPhyS: workshop on self-aware cyber-physical systems," CPS Week, April 2016, Vienna, Austria.

9. N. Capodieci, E. Hart, and G. Cabri, "Designing self-aware adaptive systems: From autonomic computing to cognitive immune networks," in *Self-Adaptation and Self-Organizing Systems Workshops (SASOW), 2013 IEEE 7th International Conference on*, September 2013, pp. 59–64.
10. S. Kounev, X. Zhu, J. O. Kephart, and M. Kwiatkowska, "Model-driven Algorithms and Architectures for Self-Aware Computing Systems (Dagstuhl Seminar 15041)," *Dagstuhl Reports*, vol. 5, no. 1, pp. 164–196, 2015. [Online]. Available: http://drops.dagstuhl.de/opus/volltexte/2015/5038
11. B. Broome, "Data-to-decisions: a transdisciplinary approach to decision support efforts at arl," *SPIE*, vol. 8389, April 2012.
12. J. O. Kephart and D. M. Chess, "The vision of autonomic computing," *Computer*, vol. 36, no. 1, pp. 41–50, 2003.
13. A. Y. Zomaya, Ed., *Handbook of Nature-Inspired and Innovative Computing.* Springer, 2006.
14. C. Müller-Schloer, H. Schmeck, and T. Ungerer, Eds., *Organic Computing A Paradigm Shift for Complex Systems.* Birkhauser, 2011.
15. M. T. Higuera-Toledano, U. Brinkschulte, and A. Rettberg, Eds., *Self-Organization in Embedded Real-Time Systems.* Springer, 2013.
16. B. Cheng, R. de Lemos, P. Inverardi, and J. Magee, Eds., *Software Engineering for Self-Adaptive Systems*, ser. Programming and Software Engineering. Springer, 2009.
17. D. Vernon, G. Metta, and G. Sandini, "A survey of artificial cognitive systems: Implications for the autonomous development of mental capabilities in computational agents," *Evolutionary Computation, IEEE Transactions on*, vol. 11, no. 2, pp. 151–180, April 2007.
18. P. R. Lewis, M. Platzner, B. Rinner, J. Torresen, and X. Yao, Eds., *Self-Aware Computing Systems: An Engineering Approach.* Springer, 2016.
19. N. Dutt, A. Jantsch, and S. Sarma, "Towards Smart Embedded Systems: A Self-Aware System-on-Chip Perspective," *ACM Transactions on Embedded Computing Systems, Special Issue on Innovative Design Methods for Smart Embedded Systems*, 2016, invited.
20. ——, "Self-Aware Cyber-Physical Systems-on-Chip," in *Proceedings of the International Conference for Computer Aided Design*, Austin, Texas, USA, November 2015, invited. [Online]. Available: http://jantsch.se/AxelJantsch/papers/2015/NikDutt-ICCAD.pdf
21. "Vital signs monitoring devices market: Increasing usage in home care settings and sports industry fuelling demand: Global industry analysis and opportunity assessment 2015-2025," London, UK, October 2015. [Online]. Available: http://www.futuremarketinsights.com/reports/vital-signs-monitoring-devices-market
22. A. Anzanpour, I. Azimi, M. Götzinger, A. M. Rahmani, N. TaheriNejad, P. Liljeberg, A. Jantsch, and N. Dutt, "Self-awareness in remote health monitoring systems using wearable electronics," in *Proceedings of Design and Test Europe Conference (DATE)*, Lausanne, Switzerland, March 2017.
23. R. Morgan, F. Williams, and M. Wright, "An early warning scoring system for detecting developing critical illness," *Clinical Intensive Care*, vol. 8, no. 2, 1997.

24. J. McGaughey, F. Alderdice, R. Fowler, A. Kapila, A. Mayhew, and M. Moutray, "Outreach and early warning systems (EWS) for the prevention of intensive care admission and death of critically ill adult patients on general hospital wards," *Cochrane Database of Systematic Reviews*, vol. 18, no. 3, July 2007.
25. D. Georgaka, M. Mparmparousi, and M. Vitos, "Early warning systems," *Hospital Chronicles*, vol. 7, no. 1, pp. 37–43, 2012.
26. A. Anzanpour, A. M. Rahmani, P. Liljeberg, and H. Tenhunen, "Context-aware early warning system for in-home healthcare using internet-of-things," in *Proceedings of the International Conference on IoT Technologies for HealthCare (HealthyIoT15)*, ser. LNICST. Italy: Springer, 2015.
27. M. Götzinger, N. Taherinejad, A. M. Rahmani, P. Liljeberg, A. Jantsch, and H. Tenhunen, "Enhancing the early warning score system using data confidence," in *Proceedings of the 6th International Conference on Wireless Mobile Communication and Healthcare (MobiHealth)*, Milano, Italy, November 2016. [Online]. Available: http://jantsch.se/AxelJantsch/papers/2016/MaxGoetzinger-MobiHealth.pdf
28. "Important Facts about Falls," 2016. [Online]. Available: http://www.cdc.gov/homeandrecreationalsafety/falls/adultfalls.html
29. I.-M. Lee *et al.*, "Effect of physical inactivity on major non-communicable diseases worldwide: an analysis of burden of disease and life expectancy," *The Lancet*, vol. 380, no. 9838, pp. 219–229, 2012.
30. G. Sprint and D. J. Cook, "Using smart homes to detect and analyze health events," *Computer*, pp. 29–37, November 2016.
31. R. Igual, C. Medrano, and I. Plaza, "Challenges, issues and trends in fall detection systems," *BioMedical Engineering Online*, vol. 12, no. 1, 2013, article 66.
32. L. Dasenbrock, A. Heinks, M. Schwenk, and J. Bauer, "Technology-based measurements for screening, monitoring and preventing frailty," *Zeitschrift für Gerontologie und Geriatrie*, vol. 49, no. 7, p. 581595, 2016.
33. A. Tessier, M.-D. Beaulieu, C. Mcginn, and R. Latulippe, "Effectiveness of reablement: A systematic review," vol. 11, no. 4, pp. 49–59, 2016.
34. S. Billinger, R. Arena, J. Bernhardt *et al.*, "Physical activity and exercise recommendations for stroke survivors," *Stroke*, vol. 45, no. 8, pp. 2532–2553, August 2014.
35. M. S. Kuster, "Exercise recommendations after total joint replacement," *Sports medicine*, vol. 32, no. 7, pp. 433–445, 2002.
36. G. D. Abowd and A. K. Dey, "Towards a better understanding of context and context-awareness," 1999.
37. J.-P. Vasseur and A. Dunkels, *Interconnecting Smart Objects with IP: The Next Internet.* Morgan Kaufmann Publishers Inc., 2010.
38. P. Harrington, *Machine Learning in Action.* Manning Publications, 2012.
39. S. Astapov and A. Riid, "A hierarchical algorithm for moving vehicle identification based on acoustic noise analysis," in *Proceedings of*

the 19th International Conference "Mixed Design of Integrated Circuits and Systems": 19th International Conference "Mixed Design of Integrated Circuits and Systems" MIXDES 2012, 2012, p. 467472.
40. D. Graupe, *Deep Learning Neural Networks. Design and Case Studies.* World Scientific Publishing Co. Pte. Ltd., 2016.
41. A. Lavin and S. Ahmad, "Evaluating real-time anomaly detection algorithms - the numenta anomaly benchmark," in *14th International Conference on Machine Learning and Applications (IEEE ICMLA)*, 2015.
42. M. A. Hassan, M. Xiao, Q. Wei, and S. Chen, "Help your mobile applications with fog computing," in *12th Annual IEEE International Conference on Sensing, Communication, and Networking - Workshops (SECON Workshops)*, 2015.
43. "Raspberry Pi - Teach, Learn, and Make with Raspberry Pi," https://www.raspberrypi.org/.
44. "PINE64 - 64-bit Single Board Computer," https://www.pine64.org/.
45. "ODROID | Hardkernel," http://www.hardkernel.com/.
46. pettitda, "Road Testing the Raspberry Pi 3 with HTM: Building the Software for 32-bit ARM," https://www.element14.com/community/groups/roadtest/blog/2016/06/07/road-testing-the-raspberry-pi-3-with-nupic, June 2016.
47. "Expanding the All Programmable SoC Portfolio," 2016. [Online]. Available: https://www.xilinx.com/products/silicon-devices/soc.html
48. J. Hawkins and S. Ahmad, "Why neurons have thousands of synapses, a theory of sequence memory in neocortex," *Frontiers in Neural Circuits*, vol. 10, no. 23, 2015.
49. V. B. Mountcastle, "The columnar organization of the neocortex," *Brain*, pp. 701–722, April 1997.
50. M. Megías, Z. Emri, T. Freund, and A. Gulyás, "Total number and distribution of inhibitory and excitatory synapses on hippocampal CA1 pyramidal cells," *Neuroscience*, February 2001.
51. "TM4C12x High performance 32-bit ARM®Cortex®-M4F based MCU," http://www.ti.com/product/TM4C123GE6PM.