Cognitive and Context Aware Assistive Environments Using Future Internet Technologies

Charalampos Doukas[1], Nikos Fotiou[2], George C. Polyzos[2] and Ilias Maglogiannis[3]

[1]University of the Aegean, Samos, Greece, doukas@aegean.gr

[2]Athens University of Economics & Business, Athens, Greece, {fotiou, polyzos}@aueb.gr

[3]University of Central Greece, Lamia, Greece, imaglo@ucg.gr

Category: Long Paper

ABSTRACT

Future Internet (FI) technologies are introducing new ways of networking and cognitive data delivery. In this paper, the potential of FI based architectures for enabling the context-aware content adaptation and specialized delivery of health related information in assistive environments is investigated. The proposed system utilizes the Publish/Subscribe Internetworking (PSI) architecture, an information-oriented architecture built for the FI using the so-called "Publish/Subscribe" paradigm. Information is brought at the center of the approach, providing several advantages: flexibility, seamless information morphing and exploitation of context, access control, and security in general. In addition to an overview of the approach and its characteristics, this work also presents the implementation of a subset of an assistive environment, using Blackadder, PSI's prototype, and illustrates its potential with an emergency service scenario for the assistive healthcare domain.

Keywords

Content adaptation, context awareness, assistive information delivery, publish-subscribe networking

1. INTRODUCTION

The introduction of the pervasive healthcare paradigm has improved awareness towards the elderly, healthcare for inhabitants of remote, isolated, and underserved locations, and the need for constant medical supervision of chronic patients. In this context, advanced electronic healthcare services that were once available only at healthcare centers, are now expected to be available through a communication network anytime, anyplace and to anyone. A medical assistive environment takes advantage of pervasive and ubiquitous technologies for delivering the above services. The term "assistive" is used more generally, including not only assisting persons with known health problems, but also empowering any human to improve the quality of life by exploiting sensing and computer-based support at all levels. Thus the type of information that is exchanged in assistive environments is characterized by diversity and the need for content adaptation.

In such an environment, the promise of the Future Internet (FI) and the new concepts and architectures that it brings (such as distributed computing and ad-hoc networking, cloud computing and elasticity in resources, the Internet of Things, etc.) can fit perfectly, enabling the development of advanced assistive applications. In this work the application of FI related technologies and architectures in assistive environments, focusing on advanced networking and context awareness issues, is discussed. The proposed methodology is based on a specific clean-slate information oriented architecture, the Publish/Subscribe Internetworking (PSI) architecture. PSI is a FI architecture which utilizes information as its building block. It is based on the so-called publish/subscribe paradigm and it is designed to support effective and flexible information governance and dissemination. Information-centric architectures appear to be ideal for pervasive healthcare environments as they achieve effective information collection, provision, processing, and governance [2]. Thus, in the remainder of this paper PSI is considered as the reference architecture for the introduction of FI technologies in assistive environments. The PSI architecture was designed within the EU FP7 PSIRP[1] project and is being further developed within the EU FP7 PURSUIT[2] project.

The remainder of this paper is structured as follows: Section 2 gives an overview of adaptation and context awareness requirements in assistive environments and briefly presents various FI architectures. Section 3 details the elements of the PSI architecture and section 4 describes proposed extensions that implement the foreseen cognitive assistive environment. Section 5 presents an implementation of a subset of the proposed architecture along with an initial evaluation and, finally, section 6 presents future work and concludes the paper.

2. BACKGROUND AND RELATED WORK

2.1 Information Delivery and Context Awareness in Assistive Environments

The success of an assistive environment relies on the proper, accurate and on-time delivery of information and data exchange among the involved parties (users, operators, sensors, etc.). The strict requirements, in addition to the complexity of assistive environments (e.g., various devices, data types, user requirements, different medical personnel involved, etc.), introduces several issues that need to be addressed:

1) Assistive environments have special requirements on content and information presentation (e.g., in the case of elderly and people with vision impairment).

2) Different presentation layers exist. For instance, patient data has to be coded in different formats for transmission and decoded respectively for a medical expert to assess it. Different device specifications

[1] http://www.psirp.org

[2] http://www.fp7-pursuit.eu/

also introduce issues in data interoperability for both collecting and transmitting various contextual data. Moreover, the access network of a user may be the bottleneck of a rich data transmission session.

3) Specialized delivery in the context of reactive/proactive data transmission (e.g., in cases of emergency, etc.) is needed.

4) The transmitted information usually contains sensitive data that needs to be secured. Patient data should be protected from eavesdropping, as well as from manipulation during transmission.

The following sections introduce the concept of context awareness in assistive environments and highlight the proposed system.

2.1.1 Context Awareness

Context awareness is the capability of networking applications to be aware of the existence and characteristics of user activities and environments. In rapidly changing scenarios, such as the ones considered in the fields of mobile, pervasive, or ubiquitous computing, systems have to adapt their behavior based on the current conditions and the dynamicity of the environment they are immersed in [3]. A system is context-aware if it can extract, interpret and use context information and adapt its functionality to the current context of use. The challenge for such systems lies in dealing with the complexity of capturing, representing and processing contextual data. In assistive environments, contextual information might refer to user context (e.g., special user requirements like nutrition, user status and location, etc.), the environment context (e.g., indoor or outdoor), and the hardware context (e.g., presentation and communication devices used, network status, etc.). To capture context information, some additional sensors and/or programs are required [6]-[8].

The way context-aware applications make use of context can be categorized into the following three classes: presenting information and services, executing an application, and tagging captured data. In more details:

- Presenting information and services refers to applications that present context information to the user, use context to propose appropriate selections of actions to the user, or personalize information delivery.

- Automatically executing an application describes services or applications that trigger a command, or reconfigure the system on behalf of the user according to context changes.

- Attaching context information for later retrieval refers to applications that tag captured data with relevant context information.

One way to capture and distribute contextual data is through the use of intelligent agents. Intelligent agents can be viewed as autonomous software (or hardware) constructs that are proactively involved in achieving a predetermined task and at the same time react to their environment. According to [4], agents are capable of: performing tasks (on behalf of users or other agents), interacting with users to receive instructions and give responses, operating autonomously without direct intervention by users, including monitoring the environment and acting upon it to bring about changes, and finally showing intelligence in order to interpret monitored events and make appropriate decisions. Agents can be: proactive, in terms of being able to exhibit goal-directed behavior; reactive, being able to respond to changes of the environment, including detecting and communicating with other agents; or autonomous, making decisions and controlling their actions independently of others. Intelligent agents can be also considered as social entities, which can communicate with other agents, by using an agent-communication language in the process of carrying out their tasks.

In the context of pervasive healthcare, intelligent agents can contribute by analyzing patient and contextual information, by distributing tasks to responsible individuals, and by informing users regarding special actions and circumstances. The necessary decision-making is usually performed through data classification on the acquired patient signals and contextual information. Generated classification results can contain information concerning the status of a patient, suggested diagnosis, behavioral patterns, etc. Data classification is an important problem in a variety of engineering and scientific disciplines such as biology, psychology, medicine, marketing, computer vision, and artificial intelligence [5]. Its main objective is to classify objects into a number of categories or classes. Depending on the application, these objects can be images, signal waveforms, or any type of measurements that need to be classified. Given a specific data feature, its classification may consist of one of the following two tasks: (a) supervised classification in which the input pattern is identified as a member of a predefined class; (b) unsupervised classification in which the pattern is assigned to a hitherto unknown class.

There is a vast array of established classification techniques, utilized in assisted environments (like those in [1]), ranging from classical statistical methods, such as linear and logistic regression, to neural network and tree-based techniques (e.g., feed-forward networks, which include multilayer perceptron, Radial-Basis Function networks, Self-Organizing Map, or Kohonen-Networks [16], to the more recent Support Vector Machines). Other types of hybrid intelligent systems are neuro-fuzzy adaptive systems, which can be comprised of an adaptive fuzzy controller and a network-based predictor. More information regarding data classification techniques can be found in [5].

2.1.2 Future Internet Architectures

The current Internet architecture has evolved around the so-called end-to-end principle. The Internet was originally designed based on the need of efficiently interconnecting mainframes and minicomputers and providing efficient remote access to them. However, as the Internet evolved, users started focusing on accessing information items rather than remote machines; nevertheless, the core design of the Internet has remained unchanged. The host-centric model and the design based on the end-to-end principle of the current Internet architecture has been identified as the root cause of many of its limitations, including lack of effective mobility, multicast, and quality of service support; security and economic issues add to the problems [9]. Various add-ons, such as Network Address Translation (NATs), Mobile IP, Content Delivery Networks (CDNs), Peer-to-Peer (P2P) overlays, etc., all violate, in various ways, several aspects of the original Internet architecture in order to provide features that were not part of the original design (or the original requirements). Keeping all these in mind, various research efforts were launched in order to redesign the Internet architecture.

In this work PSI is considered as the reference architecture and it is detailed in the next section. However, alternative architectures for the FI have been proposed in corresponding literature. Content-Centric Networking/ Name-Data Networking (CCN/NDN) [18], [24] is an information-centric architecture that proposes routing based on—hierarchical—content/data names. In the specific architecture consumers ask for content by broadcasting “Interest” packets that contain the name of the content requested. Any “Data” packet whose content name has the “Interest” name as a prefix is said to satisfy this interest (request). Note that the interest/data match in this architecture can take place in any network node, making difficult the definition of access control policies that are the core of our proposed system. 4WARD/SAIL [19] is another FP7 EU funded research effort that also advocates an information-centric Internet which will enable network diversity, allowing various types of networks to co-exist and cooperate in a smooth and cost-efficient manner. The proposed architecture envisions the Internet as a network where sub-networks will be self-manageable and network paths will be active components that will be able to affect transport services. 4WARD/SAIL also intends to shed light on the socio-economic aspects of such an approach, therefore its contributions could be supportive of our work. The Data-Oriented Network Architecture (DONA) [25] and Routing on Flat Labels (ROFL) [20] designs also explore the potential of using information as the core inter-networking component. DONA proposes a new identification scheme based on flat, self-certifying identifiers as a replacement for the DNS naming resolution scheme that enables “finding” and “fetching” content. ROFL investigates the possibility of having an internetworking architecture solely based on flat identifiers, using DHTs and hierarchical DHTs. The evaluation results of ROFL indicate that this approach is feasible and can incorporate all the internetworking structures that exist in the current Internet. The PSI architecture borrows the naming concepts of these two projects, but it proposes a different way for organizing information, using scopes. Scopes are an essential component for the design and implementation of the proposed system and will be discussed below.

3. THE PSI REFERENCE ARCHITECTURE

As already mentioned, in the context of this work, PSI is utilized as FI reference architecture. The PSI architecture is a clean-slate one, taking nothing for granted (not even the standard IP protocol and IP host-based addressing), rather applying a completely different communication paradigm at all layers of the architecture: the publish/subscribe (pub/sub) paradigm. Architectures that abide by this paradigm are mainly composed of three entities: the publishers, the subscribers and an event notification service [10]. Publishers advertise and provide information (content they have available). Subscribers are information consumers, who express their interest for specific information (content) by issuing subscription messages. The event notification service is responsible for matching the advertised content with subscribers' interests and for initiating a content forwarding process from publishers towards subscribers. The pub/sub paradigm allows for endpoints decoupling, as publication and subscription operations do not need to be synchronized (not even that publication precedes subscriptions). This decoupling offers significant security advantages and facilitates the deployment of mobility and multicast mechanisms [23]. Moreover, the pub/sub paradigm allows the existence of multiple publishers for a specific publication, enabling this way caching and multi homing [11].

Information is the building block of the PSI architecture; information is everything and everything is information [12]. Each information item in PSI is identified by a unique flat identifier named as the Rendezvous Identifier (RId). Information items are organized in scopes, which are physical or logical structures hierarchically organized that are used to locate information items, as well as to control their dissemination [22].A corporate network is an example of a physical scope, whereas a (subset of members in a) social network is an example of a logical scope. Every scope is governed by dissemination policies, which dictate who can advertise information in this scope as well as who can subscribe to a scope's information items. Furthermore, every scope is identified by a unique flat identifier, named the Scope Identifier (SId), and is managed by—at least—one network entity called the Rendezvous Node (RV). All RVs are organized in a network known as the rendezvous network. The rendezvous network enables scope lookup and it can be hierarchically organized (akin to DNS) or be flat (similar to DHTs). An information item may belong to multiple scopes. In order for a publisher to advertise an information item he possesses, he has to specify the item's RId as well as the SId of the scope that will be responsible for this information item. The RId of an information item is application specific; it can be for example the result of a hash function over the item's data. Similarly, in order for a subscriber to receive an information item, a subscription message has to be issued towards the RV of the scope in which the information item belongs, while this RV becomes the so-called Rendezvous Point for the information item. Every publisher knows at least one RV, which is his default RV. So, as it is depicted in Figure 1(a), the publisher SERVER01 sends the advertisement message to its default RV that routes it to the appropriate RV. The specific procedure is illustrated in Figure 1(a), where the publisher is shown to advertise an information item to scope DE012, which is managed by RV05; it sends the advertisement message to his default RV, i.e., RV01, and it eventually reaches RV05. When the advertisement message reaches the RV05, the publication list of the scope is updated. The subscriber therefore needs to know the SId of the scope and the RId of the information item. SIds and RIds can be learned by an out-of-band mechanism, such as a search engine (or direct communication). Upon a successful matching, the RV initiates a (multicast) delivery path creation process from a publisher towards the subscriber(s).

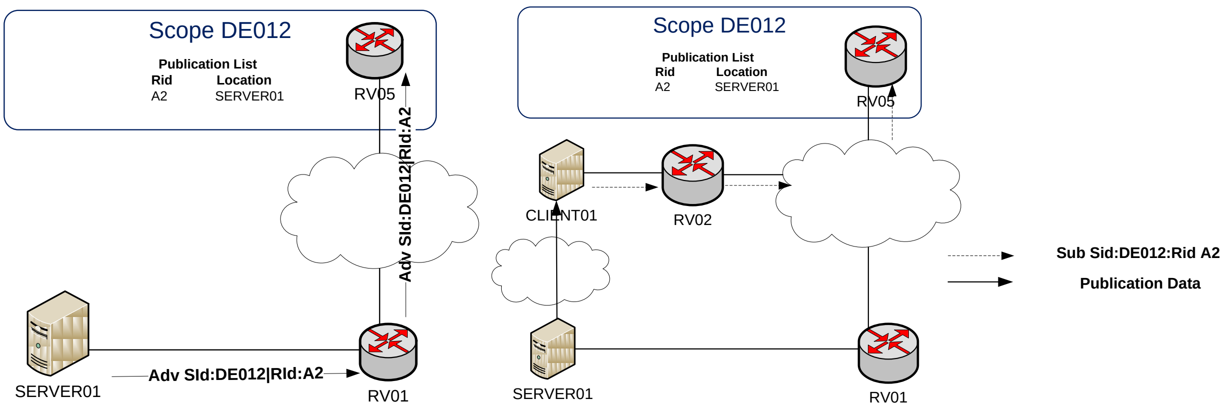

Figure 1 (a) Publication (advertisement) and (b) Subscription procedures in the PSI architecture

This process is achieved by utilizing information accumulated by a network of Topology Managers (TMs) and by taking into account subscriber requirements and possibly delivery policies. TMs are network nodes responsible for monitoring the network topology and for detecting topology changes. In order for an information item to be forwarded through a specific delivery path, a Bloom filter based structure (a space-efficient probabilistic data structure that is used to test whether an element is a member of a set) can be used, known as the zFilter [13]. zFilters contain all the links that an information item needs to traverse in order to reach its destination(s). The zFilter formation process makes it difficult for a malicious entity to manipulate a delivery path. Figure 1(b) shows a subscription example. Subscriber CLIENT01 subscribes to the already advertised information item with RId A2 in scope DE012. When the subscription message reaches RV05, a forwarding path is created between SERVER01—which is the publisher—and CLIENT01. Finally, the publication is forwarded from SERVER01 to CLIENT01.

The PSI architecture seeks to secure the information itself rather than the communication channel. Therefore, secure information can be delivered even over unsecured channels. Scopes enable information access control, as policies can be defined over who can advertise information, as well as over who can subscribe to advertised items. Information transmissions can be secured using Packet Level Authentication (PLA) mechanisms that guarantee the confidentiality and the integrity of the transmitted items. At the higher levels of the architecture, existing security solutions can be applied with minor modifications [15]. The pub/sub paradigm deployed by the PSI architecture offers a significant level of protection against (Distributed) Denial of Service ((D)DoS) attacks and spam, as there is no information flow as long as the receiver has not expressed interest on a particular piece of information, i.e., the receiver in a pub/sub architecture is able to instruct the network which pieces of information shall be delivered to it. Moreover, no information is requested from a publisher, unless the publisher has explicitly denoted its availability, i.e., unless the publisher has advertised the availability of this particular piece of information. Finally, the pub/sub paradigm enables the development of anonymization mechanisms as publication and subscription operations are decoupled. It is also able to achieve better network availability through the deployment of caching, multicast, and multi-homing.

4. THE PROPOSED ASSISTIVE ENVIRONMENT

In this section we describe an architecture that could form a foreseen assistive environment or system for cognitive health data delivery overlaid over a PSI network. The purpose of this architecture is twofold: by continually monitoring the user's health status it delivers useful information, including doctor's prescription information and reminders, medical videos and medication alerts, while in case of an emergency it provides immediate proactive information to medical personnel and alerts the responsible individuals (i.e., relatives, attending physicians). The proposed architecture is envisioned to provide new inter-networking components for the assistive system, while existing applications are expected to be compatible with this system with small modifications.

4.1 Architecture components

The proposed architecture is composed of the following modules: *Storage Devices*, *Patient Monitoring Equipment*, *Information Visualization Equipment* and *Content Transformation Nodes*. All these entities are interconnected over a PSI underlay network.

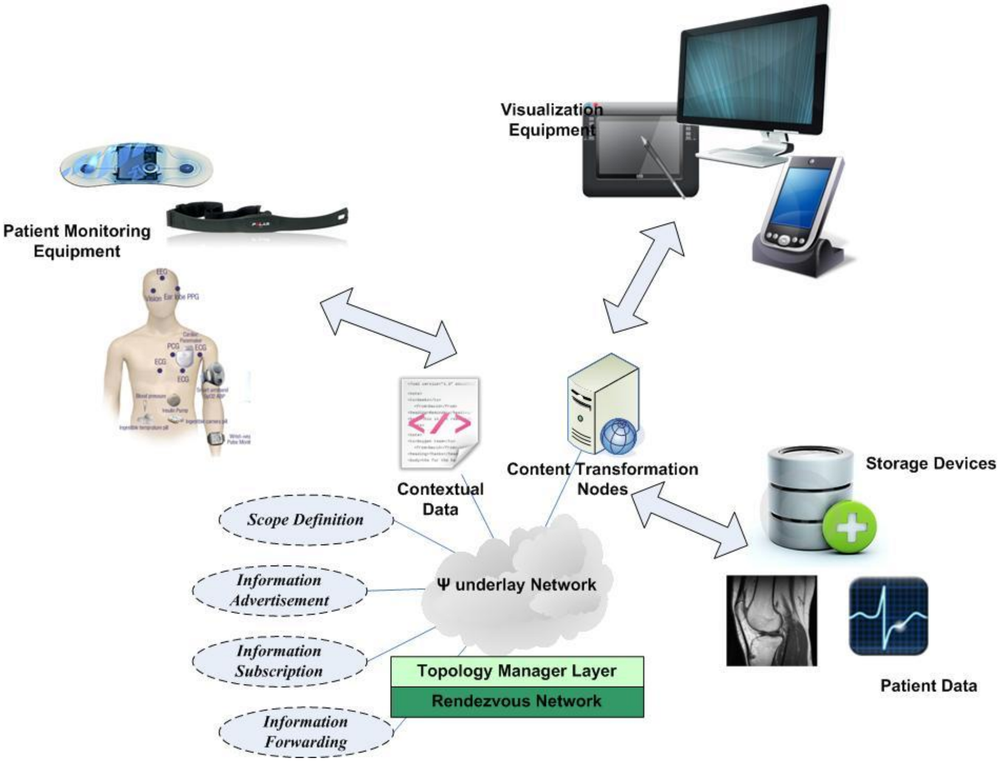

Figure 2 Main components of the proposed assistive architecture

The *Storage Devices* are network nodes that are used to store large amounts of data, including users' electronic health records, medical images, informative videos, physicians' contact information etc. These nodes have large storage capacity and achieve availability through replication.

The *Patient's Monitoring Equipment* is composed of devices that monitor a subject's health status and activity. Such devices are sensors deployed in or on the body, cameras, microphones medical equipment, etc. They are associated with the identity of a specific person using a variety of identification techniques, ranging from username-password login to smart cards or even biometrics.

The *Information Visualization Equipment* is the equipment used to view various information items such as vital signals, medical images, prescriptions and other data related to user activity and physical status. This equipment can be composed of various devices of different capabilities, such HDTVs, PC monitors, tablets and mobile phones.

The Content Transformation Nodes are network elements that are able to transform various content types into other types, for example they are able to transcode video, resize an image or read aloud text (i.e., transform text to audio).

All these nodes act as publishers and subscribers in a PSI underlay network. The PSI network is composed of the *Rendezvous Network* and the network of *Topology Managers*. The Rendezvous Network is the network responsible for the architecture's core functionality, as it enables information matching, publisher location determination and selection, creation of delivery paths, and the application of access control mechanisms. Topology Managers on the other hand are these network nodes that provide information to the Rendezvous Network regarding the current topology. Such information can be for example, that a user is connected to the architecture through a mobile network, or that a link failed.

4.2 Functions

The basic functions of the architecture are: *Scope Definition*, *Information Advertisement*, *Information Subscription* and *Information Forwarding*.

With the *Scope Definition* function, scopes are defined. Each scope has its own dissemination strategy, which is defined using an attribute-based access control scheme. Access to the information in each scope is based on user attributes and the context. Scopes can be used to advertise general-purpose information items, e.g., a scope in which information regarding a specific disease is advertised, or special purpose information items, e.g., a scope in which information regarding a specific patient's condition is advertised. For the former, general purpose access rules can be used, such as "all doctors can advertise, all users can subscribe" whereas for the latter fine grained policies have to be defined, such as "only devices of patient X can advertise, only doctor Y can subscribe." A scope's policy can be changed as needed, by taking into consideration the user's context. For example, the policy of a scope that provides the medical records of patient X can be “if the health status of patient X is normal, only attending doctor Y can subscribe; in case of emergency, the relatives of X as well as all doctors of hospital M can subscribe.” A node in the rendezvous network is chosen in order to manage a scope. Although rendezvous nodes do not store the actual content, they shall implement all these (potentially complex) policies, including (potentially frequent) updates from user devices.

With the *Information Advertisement* function the various components of the architecture advertise the information that they generate. This information is advertised to the appropriate scope, for instance to the scope that best reflects the desired dissemination strategy. Information advertisement contains an information identifier that will be used for referring to this information item, the information item's location (i.e., the location of the storage node in which this item has been stored), publisher credentials and context information, as well as metadata, such as information (MIME) type, author, size, expiration time, description, etc. The advertisement message is sent to the node of the rendezvous network that manages the desired scope. In order to make an information item available to multiple scopes, multiple advertisements have to be sent to the respective rendezvous nodes. Upon receiving an information advertisement message, the rendezvous node uses the publishers’ credentials and context information in order to validate that the publisher is eligible to advertise information under this scope.

Within the *Information Subscription* procedure, access to an information item is requested. A subscription can have many forms; it can be, for example, a subscription to a specific information item, such as “an X-Ray image,” or a subscription for a collection of items, such as “medical advice from my doctor.” Information Subscription and Information Advertisement functions are decoupled, therefore, subscriptions for content that has not yet been generated can be sent. For example, a patient may subscribe to messages that notify him that it is time to take his medicine, or a doctor may subscribe to messages that notify her that one of her patients is in an emergency situation (and when her shift ends, the subscriptions can be transferred to whoever replaces her). It should be noted here that all these notification messages are information items, also generated and advertised by the appropriate users or devices to the appropriate scope with the appropriate RId. As an example, a scope “patient X” can be defined with SId “G235” and it can be agreed that all requests for help are advertised in this scope with RId “H2EF21”; all persons that are interested in receiving “request for help” messages should send a subscription for H2EF21 to the rendezvous node responsible for managing G235 (and the authorized subscriptions will be honored). The Information Subscription message contains the identifier (Sid,Rid) of the information item to which the user wishes to subscribe (wildcards can also be used), user credentials and context information, as well as various metadata, such as desired content format, desired content type, etc. User credentials and context information are used by the rendezvous node in order to validate that the subscriber is legitimate and allowed to access the information item in question.

Information is forwarded upon a successful subscription/advertisement match, which triggers the creation of a forwarding path from information storage location towards the subscriber(s). The path creation is implemented using information provided by the Topology Managers and it has as input the subscriber's requirements and context. Subscriber requirements may include content adaptation for specific networks or devices. In this case, the forwarding path will include Content Transformation Nodes. Information forwarding can be reliable or unreliable, depending on the content type transferred as well as the subscriber's needs.

5. A USE CASE DEMONSTRATION: SMART DELIVERY OF HEALTHCARE DATA IN EMERGENCY CASES

In this section we describe the implementation of a system based on the proposed architecture in order to demonstrate its feasibility. The implementation is tested over the PSI testbed, a network of nodes across Europe, utilizing a PSI prototype node, code-named Blackadder [17][25]. The use case around the implementation is a system that monitors the vital signs and the physical context of the user and disseminates appropriate information and alerts in case of emergency, guarantying the appropriate levels of security and privacy through access control, as well as on-time delivery of critical information. In this usage scenario patients are using wearable heart-rate monitoring devices. The latter are generating ECG and heart rate signals on a regular basis, which are made available (i.e., they are advertised to a certain scope) through the PSI network. These signals are thus available to a list of (predefined) doctors, who are the patient's attending doctors (who subscribe to this item and are in scope, but the architecture makes it very easy for this list to be dynamic, based on context, with potentially elaborate conditions). Moreover, patient-monitoring equipment is capable of recognizing certain cases of emergency, such as low heartbeat rate, or a heart failure. In this case, an alarm is generated and a nearby hospital (potentially according to various preference conditions) is notified. In addition to the recent signals, data, and alarms, the system provides access to the medical history of the patient, in the form of an electronic health record, that can reside (potentially in parts) either with the user (e.g., stored in a carried personal mobile device), or at a care unit the patient has previously visited.

5.1 Components implementation

As already mentioned, our architecture is composed of Storage Devices, Patient Monitoring Equipment, Visualization Equipment and Content Transformation Nodes, all interconnected over a PSI underlay network. For our prototype implementation and tests, as basic Storage Devices and Visualization Equipment we utilize regular PCs connected to a PSI testbed. The Storage Devices contain information about the patient's context, such as the medical record, and act as local repositories for storing the acquired biosignal and healthcare data. In addition to the basic Storage Devices, the system utilizes the functionality of remote nodes that expose information such as medication instructions through Web Services. The Patient Monitoring Equipment sends directly the collected data to the Storage Devices, which in turn make this data available to the Visualization Equipment through the PSI network. The Content Transformation Nodes have been omitted from the implementation of the proposed system. In the following subsections, the Patient Monitoring Equipment, the underlay network and the API it provides are presented in more detail and it is shown how they are utilized for implementing the desired functionality.

5.1.1 Patient Monitoring Equipment

The equipment utilized in the prototype implementation for acquiring and monitoring patient biosignals is the following:

- A mobile 3-lead ECG device from Gibson Technologies[3]
- A Polar heartbeat chest strap[4] capable of continuously monitoring the patient's heartbeat.
- An Arduino microcontroller[5] for collecting the ECG signal and heartbeat. Arduino is an open-source electronics prototyping platform based on flexible, easy-to-use hardware and software. It supports a variety of extensions (shields) that provide additional functionality (e.g., collecting motion data) and networking capabilities (ZigBee, Bluetooth, WiFi, 3G/UMTS, etc.).
- An Android-based phone that forwards the collected information to the infrastructure.

The ECG interface is intended for use by the patients themselves and is easy to install. It features a pluggable interface, which can be directly connected to the Arduino board and may be used to provide more detailed information about the patient status when an abnormality is detected in the ECG signal or the heartbeat rate.

An appropriate Android application that receives the signal information and forwards it to the storage devices by invoking the appropriate Web Service calls has also been developed. In case of emergency, the application notifies a dedicated machine–acting as proxy–which in turn sends the appropriate message to the network.

5.1.2 Underlay network and API

Blackadder, the latest PSI prototype node implementation, is used in the proposed system. Several nodes across Europe are inter-linked using OpenVPN (over the existing Internet), forming in this way a large overlay network composing the PSI testbed. All nodes are identified by a unique NodeID, which is propagated throughout the network using an OSPF-like protocol. A dedicated node acts as the Topology Manager (TM) of the testbed and is responsible for monitoring the network graph and generating network paths (through several nodes at the PSI level). A separate node acts as the Rendezvous Point (RV) of the testbed. All advertisements and subscriptions are collected at the RV. In case of an advertisement and subscription match, the RV requests from the TM to create a

[3] http://www.gssteched.com/R-ECG1.html

[4] http://www.polarusa.com

[5] http://www.arduino.cc

delivery path. It then notifies the (appropriate) publisher to send data through this path. The path creation process currently supports multicast as well as multiple publishers for the same publication. In the latter case the TM chooses to use the publisher which is closer (as determined by a network metric, such as delay, or hop count) to the subscriber.

Blackadder is a piece of software that provides the basic functionality for enabling advertisement (publication) of information, subscription to information and notifications for subscriptions. Blackadder has been developed for the Linux OS using the Click modular router[6] and its source code is already publically available[7]. Blackadder provides an API that can be used to build complex applications, so it can be regarded as the basic building block of applications running over PSI.

In this work we have extended this API to support the following functions[8]

- publish_scope(scope_id,payload). This function creates a scope identified by the scope_id. The public key of the patient is used as scope_id. The payload contains the following elements: the access control definition file, as well as a digital signature over for the concatenation of scope_id and access control definition file. This signature is generated using the patient's private key.
- publish_info(scope_id,info_id,payload). This function advertises an information item, which can be, for example, the identifier of an ECG file or a medical image. The item is advertised under the scope scope_id, which means all access rules defined in this scope are applied to this item. This item is identified by the info_id, which is the result of a hash function over its data. The payload may include metadata such as the type of the published data, information about its creation date, device used to generate it, etc. Scope_id, info_id and payload are digitally signed using the patient’s private key.
- publish_data(scope_id,info_id,payload). This function publishes (forwards) the data connected with an already advertised info_id. The payload includes the following: the content to be published and a digital signature over the data, the scope_id and the info_id. A publish_data() API call is always invoked after the respective publish_info() call, i.e., an information item should firstly be advertised—using publish_info().
- subscribe_scope(scope_id,payload). With this function the subscribers subscribe to a scope. The payload contains the subscriber credentials, a digital signature over the credentials and the scope_id. With this function the subscriber will learn all the items that are advertised in that particular scope. Moreover, this function creates persistent state, i.e., the subscriber will be notified about all future items advertised in

[6] http://www.read.cs.ucla.edu/click/click

[7] https://github.com/fp7-pursuit/blackadder

[8] Each function call requires additional parameters—such as payload size—which are omitted for clarity.

that scope. Nevertheless, in case of subscriber disconnection, these notifications will be lost. Separate mechanisms are provided in order to handle subscriber disconnection and mobility (e.g., a subscriber could subscribe from multiple locations). Subscriptions to scopes never expire; therefore, a leasing/renew mechanism is not necessary.

- subscribe_info(scope_id,info_id,payload). This function enables subscription to a publication already advertised in the scope_id. The payload contains the subscriber's credentials, a digital signature over the credentials, the scope_id and the info_id. Since in the assistive architecture all information items are mutable, subscription messages do not create persistent state; every time a subscriber receives the desired information item, the state that the subscription message has created is erased.

For the publish_scope() and publish_info functions() the relative unpublish function exists. Similarly, for each subscribe_scope() and subscribe_info() functions the relative unsubscribe function exists.

An interesting feature of the proposed API–compared to the core Blackadder API–is that it does not introduce any constraint on the length and the form of the identifiers. This is particularly helpful as it allows the usage of (encryption) public keys as identifiers, which–as will be discussed in the next section–enables the secure communication of the various entities of the system.

Using this extended API two higher layer RV nodes have been built: the first handles the scopes that are created by the patients and the second handles the scopes used for advertising emergency events (further elaboration on these two kinds of scopes is provided in the next section). The higher layer RVs use the Blackadder's topology management function in order to create paths among arbitrary nodes. It is noteworthy, however, that the prototype, in its current form, cannot handle dynamic topologies; the network graph has to be fed as a configuration file to Blackadder's TM.

5.1.3 Functionality of the System

In the described implementation, two types of scopes are distinguished: the first type is the patient's private scope, where patient's data is advertised and the second type is a general-purpose scope, where emergency events are advertised. All patients have a public/private key pair. Each patient's scope has his public key as id–denoted as Patient_PK–whereas the emergency scope has a well-known pre-configured ID, which for the remainder of this paper is denoted as EVENT_SID[9]. Moreover, two types of identities are used: the personal identity, which is assigned to a single user, e.g., Bob, and the role identity which is assigned (in general) to a group of users, e.g., doctors of ER. A public/private key pair identifies each single user and each specific role. The rendezvous point is provided as a service by the network. Users and devices do not have to be aware of the rendezvous

[9] It should be noted here that other general purpose scopes can be created for other events. Moreover, different types of emergency events can have their own scopes.

implementation details, but they should be aware of its public key—denoted as RVService_PK—which is the key with which they should encrypt the payload of all the messages that are sent to the rendezvous system. In the current implementation, storage devices are owned by the patient, thus the patient's private key is known to them.

Initially each patient creates his private scope by calling the publish_scope(Patient_PK, payload) function. In the payload, for the current static implementation, the patient’s agent inserts two lists of public keys. The first list is the list of persons that have access to his medical data, whereas the second list is the list of persons and/or roles that have access to his medical data in case of emergency. In case a patient wants to modify the access control lists, he simply has to call again the same function with the new access control lists. The payload is encrypted with the RVService_PK

Doctors have to subscribe to their patients' scope (through their terminal devices). This is achieved by executing the subscribe_scope(Patient_PK, payload) function, once for each patient, providing their credentials in the payload. If a doctor wants to stop receiving data from a specific patient, the unsubscribe_scope(Patient_PK, payload) function can be used. The payload is encrypted with the RVService_PK.

Every time a storage device receives new data, it calculates its ID by applying a hash function. Then, it executes the publish_info(Patient_PK, info_id, payload) function. Almost immediately, the doctors in the access control list who have subscribed to the patient's scope will receive a notification that new data is available. If a doctor is interested in receiving the advertised data she (i.e., her software agent on her terminal device) uses the subscribe_info(Patient_PK, info_id, payload) function. As a next step, the storage device receives a notification that there are available subscribers. Upon receiving such notification, the storage device sends the data by executing only once the publish_data(Patient_PK, info_id, payload) function and all the doctors that are in the access control list and have subscribed to this data will receive it. This is achieved because the TM has already calculated and configured the delivery tree from the storage devices to doctor's visualization equipment. Every time a storage device erases some data, it executes the unpublish_data(Patient_PK,info_id,payload). Figure 3 shows an example of data exchange under normal circumstances[10]. In this example a patient (Patient_A) wears a heartbeat chest strap. This strap generates ECG signals, which are sent to two out of the three storage devices using HTTP_POST. If a scope for the patient has not been created, one of the storage devices executes the publish_scope function. All the storage devices that receive the ECG data advertise it to the PSI underlay network. At the same time a doctor, Doctor1, subscribes from his terminal device to the patient's scope. The PSI underlay network assures that Doctor1 is allowed to subscribe to this scope and notifies the doctor's terminal about the new ECG signal. As a next step, the doctor subscribes to receive this specific data. The PSI underlay network through the TM calculates a path from the storage device, which is closer to the doctor's terminal, which in this case is the Storage Device 3. The underlay network notifies Storage Device 3, informing it of the path

[10] For clarity, payloads have been omitted.

identifier. Finally, Storage Device 3 publishes the ECG, which is forwarded to the doctor’s terminal through the created path.

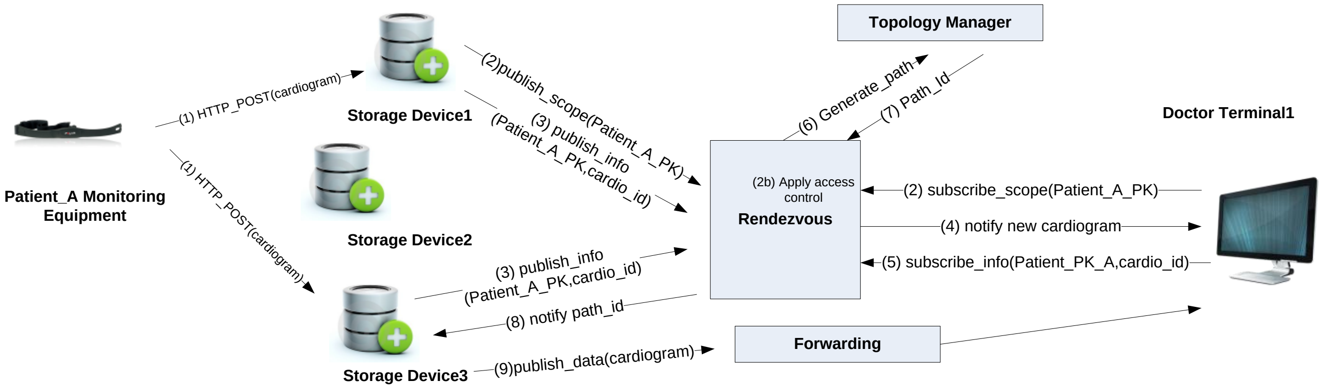
Topology Manager
(1) HTTP_POST(cardiogram)
Storage Device1
(2)publish_scope(Patient_A_PK)
(3) publish_info (Patient_A_PK,cardio_id)
(6) Generate_path
(7) Path_Id
Doctor Terminal1
Patient_A Monitoring Equipment
(1) HTTP_POST(cardiogram)
Storage Device2
(2b) Apply access control
Rendezvous
(2) subscribe_scope(Patient_A_PK)
(4) notify new cardiogram
(5) subscribe_info(Patient_PK_A,cardio_id)
(3) publish_info (Patient_A_PK,cardio_id)
(8) notify path_id
Storage Device3
(9)publish_data(cardiogram)
Forwarding

Figure 3 An example of data exchange under normal conditions

In case of emergency, the patient advertises an alert in the emergency scope using as info_id his public key, therefore, the following function is executed: publish_info(EVENT_SID,Patient_PK,payload). All the rendezvous nodes that are managing patient data will receive this notification. Then, the rendezvous nodes will relax the access control policies that control patient data and will notify the subscribers that are included in the emergency access control list about the data that is available in the patient's scope, providing at the same time information about the emergency event. The subscribers will examine this data and will subscribe to the data item(s) that are necessary in order to handle this emergency event. When the patient issues an unpublish_info(EVENT_SID,Patient_PK,payload), the users included in the emergency access control list will not have access to this patient's data any more. Figure 4 gives an example of emergency case scenario. Patient_A, carries a heartbeat chest strap which captures bio-signals and detects patient status. Initially, the rendezvous node that manages the patient's biosignals subscribes to a special scope used for emergency notifications. Moreover, the healthcare unit (i.e., Hospital1) that is responsible for Patient_A in case of emergency subscribes to receive his medical data. In this example, the monitoring equipment detects in the captured ECG signals a heart failure, therefore, it calls a Web service located in Storage Device1. This device advertises an event in the emergency scope, which eventually notifies the rendezvous system that Patient_A is experiencing a heart attack. This notification results in the relaxation of the access control policies and the notification of the Hospital Terminal1. The notification message contains the Patient_A identifier, a code corresponding to the type of emergency, as well as a list of the IDs of the advertised data. In this case, the doctor monitoring this terminal decides to subscribe to a particular ECG file, cardio_1. This subscription leads to the creation of a forwarding path from Storage Device 2, which is the storage location for this specific data, to the terminal.

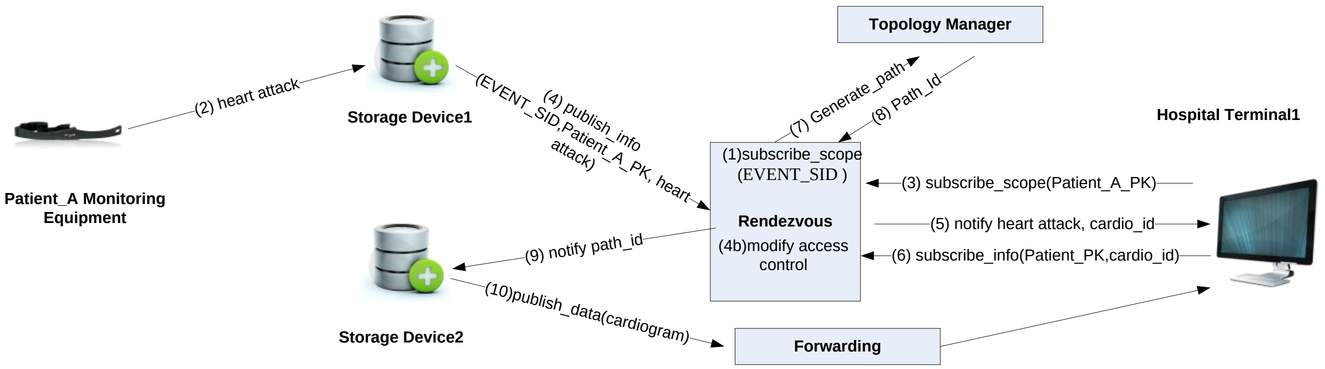
Topology Manager
Storage Device1
(2) heart attack
(4) publish_info
(EVENT_SID,Patient_A_PK, heart attack)
(7) Generate_path
(8) Path_Id
Hospital Terminal1
(1)subscribe_scope (EVENT_SID)
(3) subscribe_scope(Patient_A_PK)
Patient_A Monitoring Equipment
Rendezvous
(5) notify heart attack, cardio_id
(4b)modify access control
(6) subscribe_info(Patient_PK,cardio_id)
(9) notify path_id
(10)publish_data(cardiogram)
Storage Device2
Forwarding

Figure 4 Messages exchanged in case of emergency

5.2 Initial Evaluation of the System

In this section we evaluate the proposed system, mainly from a qualitative perspective, in terms of performance, extensibility, and security.

5.2.1 Scalability and Performance

The developed prototype allows the deployment of multiple storage points and supports replication and multicast. Each patient is not limited to a single storage point and multiple devices can be used for the same patient. Being location independent, the proper operation of the developed architecture depends only on the IDs assigned to the information items. The generated ECG files and vital data can be replicated to multiple storage devices deployed in various parts of the network. When the TM is requested to generate paths between storage devices (which act as publishers) and doctor's visualization equipment (which act as subscribers), the storage device that is closer to the doctor's terminal, in terms of (some) network distance, will be selected. Moreover, if multiple doctors request the same item (and this is a typical case as all the doctors of a patient may have subscribed to receive everything that is advertised in the patient's scope), the TM will create a multicast delivery tree from the appropriate storage device towards the physician's visualization equipment. Every request in the system concerns a unique object identifier and not a network location. Requests on object identifiers facilitate the deployment of caches.

A system based on the current Internet architecture would require CDN-like add-on solutions in order to allow content replication and optimal selection of content sources, since the current Internet architecture does not natively provide caching, nor multi-homing. Moreover, since the current Internet architecture does not support multicast in an effective way, the application developers should also be concerned with adding application-layer multicast support in their systems. In the current Internet all operations refer to network locations rather than objects. Therefore, in order to achieve network caching, application layer, or deep packet inspection mechanisms have to be deployed.

5.2.2 Extensibility

New devices and users can be added to the developed system in an easy and straightforward manner. Any device equipped with the proper API can interact with the PSI network, with little or no configuration. It should be noted that this API is not system-specific; on the contrary, it is a general purpose API provided by the underlay architecture, which will be used by other systems as well. This way, interoperability across applications is better supported. In the current Internet, system specific mechanisms, such as Web services, have to be developed in order to support device and application interoperability. Whenever a new user is added to the system there is no need for configuration in the rendezvous system, which is responsible for applying the access control policies. Patients wishing to enter the system have simply to create a new scope, which uses their public key as ID. The rendezvous system is able to determine that a new patient is a valid user by verifying their public key using the digital signature of the packet. In order for doctors to enter the system, they have to simply send a subscription message with their public key. The rendezvous point is able to determine the validity of the subscription by verifying the public key used in order to sign the message. If doctors are included in any of the patient's access control lists, they will receive the appropriate data. Forwarding in PSI is based on source routing; therefore a rendezvous node can control the nodes through which data will be transferred. Source routing makes the deployment of special purpose nodes, such as content transcoders, easier. When such nodes are introduced into the system, a rendezvous node can be simply configured to direct traffic through these nodes if necessary. Since PSI is location independent, in contrast to the current Internet, a patient or a doctor is not bound to a particular network location. So, if a patient decides for example to provide their medical records from a different storage point, the overall system operation will not be affected at all, or if a doctor changes office and logs on to the system from another terminal, their session will be transferred to the new terminal. In the current Internet architecture this cannot be easily achieved; either all endpoints will have to be reconfigured in order to communicate with the new endpoint, or indirection mechanisms−such as DNS−will have to be used (and many times in ways not intended).

5.2.3 Security

The exchanged messages are digitally signed with the sender's key, therefore, their integrity can be verified. Moreover, by having all the messages signed, effective accountability mechanisms can be deployed, at least at the application layer. Data confidentiality is achieved by encrypting the payload using the RVService_PK. All the notifications sent from the rendezvous system to the publishers are encrypted using their public key. Each item is identified by the public key of the owner, along with the result of a hash function applied over its data. Upon receiving an information item, a subscriber is able to verify its provenance (by evaluating the digital signature of the message using the publisher's public key) and its relevance (by evaluating the hash of the received content). Moreover, since data is encrypted, it can be transmitted over an unsecured channel. Information item verification can also be done by in-network mechanisms. In case of multicasting, a different solution should be considered. In this case the rendezvous system creates a group communication key, which is sent to the subscribers and to the publisher. This solution obviously requires a side channel in order to send the key.

Access to patient data is assured using access control. Provided that the rendezvous system is reliable, only legitimate users can access patient files. Moreover, removal (revocation) of a user from an access control list, or addition of a new user, is straightforward as a single message is required (containing the new access control list).

On the other hand, security in the current Internet is closely tied to data location (e.g., security certificates are bound to a specific URL). Moreover, security in the current Internet mainly concerns securing the underlying communication channel (e.g., using HTTP over SSL/TLS in order to provide secure HTTP). The proposed architecture relaxes these limitations. In order for an application that uses the current Internet architecture to achieve the security properties of the system, application specific mechanisms need to be developed.

5.2.4 Cost

A concern that may arise regarding the implementation and installation of the proposed system is its cost. Nevertheless, providing that the underlay FI architecture is deployed, the cost for adapting existing equipment and applications is expected to be low. In our implementation, proxies have been used, translating HTTP messages sent from the patient equipment into PSI specific API calls. Such proxies could be used in a real, incremental deployment. However, since proxies may introduce performance penalties, the gradual modification of the communication protocol of critical equipment should be considered.

The previously analyzed evaluation is summarized in Table 1.

	PSI based assistive system	Current Internet based assistive system
Scalability and Performance	• Support for multi-homing • Support for multicast • Caching can be easily deployed	• Application layer solutions required for effective data transmission (such as CDNs, application layer multicast) • Deep packet inspection for caching, or application layer caching
Extensibility	• Better support for interoperability • Source routing facilitates the introduction of special purpose in-network nodes • Location independent mechanisms facilitate the mobility of the various stakeholders	• System specific mechanisms—such as web services—have to be created in order to support system inter-operability • Separate indirection mechanisms are required in order to support user mobility
Security	• Information-oriented security mechanisms • Information can be transmitted over unsecured channels • Self-certifying names on information items	• Location based mechanisms • Application layer solutions have to be deployed in order to secure content (such as encryption-key exchange mechanisms)

Table 1 Comparison between the proposed system and a similar system that uses current Internet technologies

6. DISCUSSION AND CONCLUSIONS

Context-aware assistive environments face difficulties due to limitations of the current Internet technologies. Such systems demand complex information manipulation and effective security mechanisms, which currently have to be provided with add-on solutions. This is necessary as users utilize various devices for producing and retrieving information and in particular because the generated content includes sensitive data, which has to be properly secured and kept private.

In this paper, a conceptual solution is presented for a context-aware assistive environment based on a Future Internet architecture: the PSI architecture. In addition, it is shown that by designing an architecture around information items and by providing functions for manipulating and securing them, pervasive health assistive solutions can be deployed more easily and effectively. Moreover, part of this conceptual architecture has been implemented by utilizing a PSI prototype. Although in small scale, the implementation gives a hint about the scalability, extensibility, and security features of such architectures and systems.

As the PSI prototype evolves, it is expected that functionality currently implement by add-on solutions will be incorporated into the system's core design. In the current prototype a single machine performs the rendezvous function. Future releases of the prototype will include distributed rendezvous functionality enabling more scalable handling of publications and subscriptions. Currently, the topology manager of the prototype creates paths among end-points without taking into consideration user preferences. It is expected that in future releases it will be possible for a publisher or a subscriber to include in their requests policy requirements, e.g., nodes through which data should be sent or nodes and networks to be avoided. This new feature will also enable the deployment and exploitation of content transformation nodes: subscribers or publishers requiring content transformation will request the delivery paths to include these nodes.

In the adopted security approach, public keys were used to define the desired access control policies. Future work in this domain includes the possibility of the definition of access control policies using additional attributes. Such attributes can include advanced location and context aware policies. For example, policies such as "in case of emergency, all nearby hospitals must be notified," should be possible to be easily implemented (and in very dynamic fashion, e.g., considering not only the location of the emergency, but also time of day, traffic conditions, other emergencies, etc.). The API produced during this implementation allows the creation of a single scope per patient. Future versions will allow the creation of a hierarchy of scopes, allowing for better organization, containment, and access to information.

It should be noted that currently, patient monitoring devices are communicating directly with the storage devices owned by the patient using HTTP. However, the intention of the architecture and future work plans include allowing the devices to interact with the core PSI network. If user devices are directly integrated into the network, then storage can be regarded as a network service, as cloud storage is provided today, or in-network caches will be in the near future.

Finally, future work plans include the implementation of an overlay[11] version of the system, which will allow co-existence with IP networks and thus a faster path to adoption through backward compatibility. This overlay version will also enable extensive, large-scale tests with real users and data.

7. REFERENCES

[1] Doukas C, Maglogiannis I (2011) An assistive environment for improving human safety utilizing advanced sound and motion data classification. Universal Access in the Information Society 10:217-228

[2] Trossen D, Pavel D, Guild K, Bacon J, Singh J (2010) Information-centric Pervasive Healthcare Platforms. In: Proceedings 4th International Conference on Pervasive Computing Technologies for Healthcare (Pervasive Health)

[3] Khedo KK (2006) Context-Aware Systems for Mobile and Ubiquitous Networks. In: Proceedings International Conference on Networking, International Conference on Systems and International Conference on Mobile Communications and Learning Technologies (ICN/ICONS/MCL)

[4] Fox J, Beveridge M, Glasspool D (2003) Understanding intelligent agents: analysis and synthesis. AI Communications. IOS Press 16(3):139-152

[5] Zhai JH, Zhang SF, Wang XZ (2006) An Overview of Pattern Classification Methodologies. In: Proceedings of the Fifth International Conference on Machine Learning and Cybernetics 3222 – 3227

[6] Doukas C, Maglogiannis I, Kormentzas G (2006) Advanced Telemedicine Services through Context-aware Medical Networks. In: Proceedings International IEEE EMBS Special Topic Conference on Information Technology Applications in Biomedicine

[7] Maglogiannis I (2009) Introducing intelligence in electronic healthcare systems: State of the art and future trends Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 5640 LNAI, pp. 71-90.

[8] Doukas C, Maglogiannis I, Karpouzis K (2008) Context-aware medical content adaptation through semantic representation and rules evaluation Proceedings - 3rd International Workshop on Semantic Media Adaptation and Personalization, SMAP 2008, art. no. 4724861, pp. 128-134

[9] Feldmann A (2007) Internet Clean-Slate Design: What and Why? ACM SIGCOMM Computer Communication Review 37(3):59-64

[10] Eugster PT, Felber P, Guerraoui R, Kermarrec AM (2003) The Many Faces of Publish/Subscribe, ACM Computing Surveys 23(2):114-131

[11] Katsaros K, Xylomenos G, Polyzos GC (2011) MultiCache: an Overlay Architecture for Information-Centric Networking. Computer Networks 55(4):936-947, Elsevier, special issue on 'Architectures and Protocols for the Future Internet'

[11] As in [11] and also considered as a deployment strategy for PSI in PSIRP and PURSUIT [12].

[12] Tarkoma S, ed. (2010) PSIRP deliverable 2.3, architecture definition, component descriptions, and requirements. http://www.psirp.org/

[13] Jokela P, Zahemszky A, Rothenberg CE, Arianfar S, Nikander P (2009) LIPSIN: line speed publish/subscribe inter-networking. ACM SIGCOMM Computer Communications Review 39(4): 195–206

[14] Lagutin D, Tarkoma S (2010) Cryptographic Signatures on the Network Layer–an Alternative to the ISP Data retention. In: Proceedings of IEEE Symposium on Computers and Communications, Riccione, Italy

[15] Nikander P, Marias GF (2008) Towards Understanding Pure Publish/Subscribe Cryptographic Protocols. In: Proceedings of the Sixteenth International Workshop on Security Protocols. Cambridge, England

[16] Witten IH, Frank E (2005) Data Mining: Practical machine learning tools and techniques, 2nd Edition, Morgan Kaufmann, San Francisco

[17] Kjällman J, ed (2011) PURSUIT deliverable 3.2, first lifecycle prototype implementation (d3.2). http://www.fp7-pursuit.eu

[18] Jacobson V, Mosko M, Smetters D, Garcia-Luna-Aceves JJ (2007) Content-centric networking. Whitepaper, Palo Alto Research Center

[19] SAIL Project (2011) D-3.1 (D-B.1) The Network of Information: Architecture and Application. http://www.sail-project.eu/wp-content/uploads/2011/08/SAIL_DB1_v1_0_final-Public.pdf

[20] Caesar M, Condie T, Kannan J, Lakshminarayanan K, Stoica I (2006) ROFL: routing on flat labels. ACM SIGCOMM Computer Communications Review 36(4):363-374

[21] Koponen T, Chawla M, Chun B, Ermolinskiy A, Kim KH, Shenker S, Stoica I (2007) A data-oriented (and beyond) network architecture. ACM SIGCOMM Computer Communications Review 37(4):181-192

[22] Fotiou N, Trossen D, Polyzos GC (2012) Illustrating a Publish-Subscribe Internet Architecture. Telecommunication Systems 51(4):233-245, Springer, Special Issue on 'Future Internet Services and Architectures: Trends and Visions'

[23] Xylomenos G, Vasilakos X, Tsilopoulos C, Siris VA, Polyzos GC (2012) Caching and Mobility Support in a Publish-Subscribe Internet Architecture. IEEE Communications Magazine 50(7):52-58, special issue on 'Information-Centric Networking'

[24] Jacobson V, Smetters DK, Thornton JD, Plass MF, Briggs NH, Braynard RL (2009) Networking Named Content. In: Proceedings of ACM CoNEXT

[25] Trossen D, Parisis G (2012) Designing and realizing an information-centric internet. IEEE Communications Magazine 50(7):60-67, special issue on 'Information-Centric Networking'