ICE++: Improving Security, QoS, and High Availability of Medical Cyber-Physical Systems through Mobile Edge Computing

Alberto Huertas Celdrán*, Félix J. García Clemente*, James Weimer†, and Insup Lee†
* University of Murcia
Email: alberto.huertas@um.es,fgarcia@um.es
† University of Pennsylvania
Email: weimerj@seas.upenn.edu, lee@cis.upenn.edu

***Abstract*—The disruptive vision of Medical Cyber-Physical Systems (MCPS) enables the promising next-generation of eHealth systems that are intended to interoperate efficiently, safely, and securely. Safety-critical interconnected systems that analyze patients' vital signs gathered from medical devices, infer the state of the patient's health, and start treatments issuing information to doctors and medical actuators should improve the patients' safety in a cost-efficient fashion. Despite the benefits provided by the MCPS vision, it also opens the door to critical challenges like the security and privacy, Quality of Service (QoS), and high availability of the devices composed to support the MCPS scenario. The Integrated Clinical Environment (ICE) standard is a significant step toward promoting open coordination of heterogeneous medical devices by considering the previous challenges. However, a lot of effort is still required in order to cover the whole aspects of these challenges and enable the future eHealth. In this context, we identify critical shortcomings of ICE using challenge scenarios regarding security, QoS, and high availability. According to these concerns and following the ICE standard, we propose the novel ICE++ architecture, which is oriented to the Mobile Edge Computing paradigm and combines SDN and NFV techniques to manage efficiently and automatically the MCPS elements taking into account its security, QoS, and high availability. Finally, we perform experiments that demonstrate the potential usefulness of our solution regarding the efficient and automatic management of the ICE components.**

***Index Terms*—MCPS, ICE, MEC, security, privacy, QoS, high availability**

I. Introduction

Medical Cyber-Physical Systems (MCPS) [1] refer to safety-critical interconnected systems that analyze patients' vital signs gathered from medical devices, infer the state of the patient's health, and initiate treatments issuing information to doctors or directly to medical actuators. This disruptive vision has the potential to enable in a cost-efficient way the next-generation of eHealth, which requires systems able to interoperate efficiently, safely, and securely [2].

With the goal of making possible the MCPS vision, the ASTM F2761 standard proposes a patient-centric architecture for Integrated Clinical Environments (ICE) [3] that enables the open coordination of heterogeneous medical devices and applications. The ICE framework defines a set of elements that compose the clinical environment. Among the proposed components, the most relevant are the *ICE Equipment Interfaces*, which interact with medical devices to enable their hardware and software network connection capabilities; the *ICE Supervisor*, focused on hosting medical applications that receive and control the patients' vital signs; the *ICE Network Controller*, in charge of managing the communications between the supervisor and the ICE equipment interfaces, as well as handling and maintaining the discovery of medical devices and their information; the *Data Logger*, focused on troubleshooting and forensic analysis; and *External Interfaces*, which enable the communication with external hospital resources such as Electronic Health Records (EHR). Fig. 1 shows the elements composing the ICE framework and their communications.

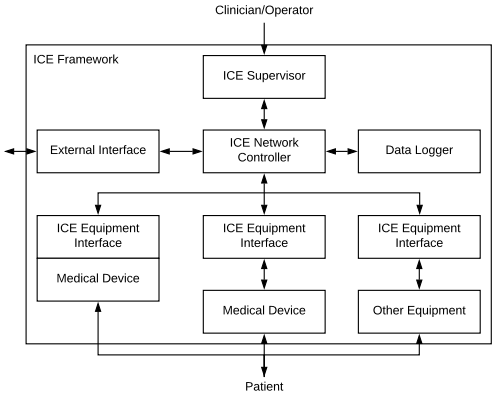

Fig. 1. Elements composing the ICE framework

Any implementation of the ICE framework should consider several critical pillars to enable the MCPS vision [2]. The majority of these pillars are a work in progress and open challenges. Among the existing challenges, we highlight:

- *Security and privacy*. Since medical systems contribute to the care received by patients and manage their private information, any ICE implementation must be secure and robust. The authors of [4] show that the healthcare industry lags behind in security. Healthcare solutions

978-1-5090-0223-8/16/$31.00 © 2016 IEEE

978-1-5386-4294-8/18/$31.00 ©2018 IEEE

should clearly define their cybersecurity issues and establish clear procedures for managing attacks and data breaches [5]. In this sense, ICE solutions do not have enough capabilities to detect and mitigate cyber-attacks. This is a critical shortcoming because cyber-attacks affecting the security and privacy of healthcare and ICE components can endanger the patient's life.

- *Quality of Service (QoS)* through the scalability, efficiency and performance. Healthcare solutions should exhibit a great capacity to reuse and optimize its resources even in unexpected situations. Current healthcare systems generate large amounts of heterogeneous real-time data. Emergence situations, where the number of medical devices and patients is higher, increase the volume of data making even more challenging its management within acceptable periods of time [6]. In this context, current ICE solutions are not able to automatically manage the computational resources and then make decisions to ensure the best service to the patients. This is an important drawback because ICE solutions can be overloaded by provoking unacceptable delays in the patients' treatments.
- *High Availability* of the ICE framework. The Food and Drug Administration (FDA) receives thousands of reports every year about suspected device-associated deaths, serious injuries, and malfunctions [7]. The Medical Device Reporting (MDR) is one of the tools used by the FDA to detect potential device-related safety issues [7]. In this sense, medical components that fail or have problems should be seamlessly replaced by redundant data sources or other hardware devices if they are available. Current ICE solutions do not provide autonomous mechanisms to deploy, manage, and dismantle the ICE components when they suffer failures. High Availability is critical because if one of the ICE components fail due to misconfiguration, misuse, hardware failures, or natural catastrophes, the patients can be injured.

Although the combination of Mobile Edge Computing (MEC) [8] with new technologies as Network Function Virtualization (NFV) and Software Defined Networking (SDN) [9] is being successfully used in 5G Mobile Networks, we believe that these new technologies can also provide solutions to other scenarios. In these sense, our proposal is based on the novelty of combining NFV and SDN with MCPS to provide a flexible, efficient, and fault-tolerant platform suitable for facing the previous challenges. Specifically, the MEC paradigm will enable the management of the components defined by the ICE framework in the edge of the network, which is a critical aspect in order to ensure the low-latency associated to the QoS [10]. On the other hand, NFV techniques have changed the vision of the networking paradigm by allowing the flexible and efficient control and deployment of virtual network infrastructure and services. This aspect will enable the high availability, scalability, and efficiency required by MCPS through the deployment and control of ICE components as well as the network infrastructure. Finally, the SDN paradigm separates the data and control planes of the network to enable the real-time management of the network communications. It will guarantee the communication management of the ICE framework in real-time and on-demand.

By taking into account the potential of the previous technologies, this paper has three main contributions:

1) A novel extended ICE architecture, called ICE++, that combines the MEC paradigm with the SDN and NFV techniques to improve the security, privacy, QoS, and high availability challenges of current ICE solutions. The combination of NFV, MEC, and SDN allows our architecture to deploy and control in a flexible and efficient way the components making up the ICE framework as well as its communications at the edge of the network.
2) A systematic comparison between the ICE and ICE++ frameworks using realistic clinical scenarios for security, QoS, and high availability.
3) A performance evaluation using ontologies and semantic reasoners that illustrates the viability of our solution when it makes decisions to manage the previous challenges. Specifically, for a large hospital in Pennsylvania with 650 beds, our solution detects and reacts to anomaly situations in less than three seconds.

The remainder of the paper is structured as follows. Section II discusses some related work on security, privacy, QoS, and high availability challenges of ICE solutions. Section III shows the components forming the proposed architecture. Section IV shows three realistic scenarios with the concerns of current ICE implementations. Section V shows the experiments performed to demonstrate the usefulness of our solution. Finally, conclusions and future work are drawn in Section VI.

II. Related Work

Historically, medical devices have been developed as standalone systems without communication capabilities. However, the MCPS vision is emerging to provide interoperability, safety, and security to clinical environments. Nowadays, OpenICE [11] is a commonly adopted implementation of the ICE framework. OpenICE is a distributed patient-centric architecture that implements the components defined by the ICE framework. On the one hand, the equipment interfaces can run on computers with limited resources (e.g. beaglebone black, raspberry pi, etc.), which are physically attached to medical devices to provide network capabilities [12]. On the other hand, the communications between the interfaces and the supervisor are managed by the external DDS middleware [13], which covers partially the QoS and Security of OpenICE. In spite of that, there is still an important gap to work on improving the main challenges of MCPS.

In [14] is presented a solution to protect the communications of the ICE framework through the security mechanisms provided by the OMG Data Distribution Service (DDS) standard [13]. Despite the important outputs of this proposal, it is not clear how DDS is able to mitigate or at least improve DoS attacks. Another work, oriented to protect the security and privacy of solutions based on ICE, is proposed in [15]. The

authors of this work propose a cloud-based secure logger that receives the information sensed by ICE interfaces attached to medical devices. This solution is effective against replay, injection, and eavesdropping attacks. However, any behavior that does not lead to altering of messages is not detected. In [16], it is designed and implemented an authentication framework for ICE-compliant interoperable medical systems. The proposed framework is composed of three layers, allowing it to fit in the variety of authentication requirements from different ICE entities and networking middlewares. The experiments demonstrate that the proposed authentication framework protects to replacement and impersonation attacks.

OpenICE implements the DDS middleware, which provides QoS through the next parameters: *liveliness*, *durability*,*history*, and *reliability*. In comparison to OpenICE the work proposed in [17] presents a set of communication patterns that can facilitate reliable composition of medical systems. The proposed patterns intent to enable unambiguous description of communication between devices and applications, including QoS requirements. In [18], the authors enhance the capabilities of the ICE framework with online service composition and reconfiguration of medical systems. A service-based eHealth for remote monitoring of patients is implemented to validate the improved capacity of ICE to support dynamic reconfiguration of application. In [19], it is developed a real-time architecture that considers the RTPS middleware functions into healthcare systems. Furthermore, they evaluate the developed system over wired and wireless channels in terms of throughput and delay.

High availability is another essential challenge of the ICE framework. In [20], it is proposed a solution that considers common hardware failures and malicious attacks to make the ICE Supervisor a trusted entity. Specifically, the proposed lightweight mechanisms lie on the replication of the supervisor component. Space redundancy is necessary to protect the supervisor from partial hardware failures while offering continuous high availability. The availability of the rest of elements composing the ICE framework is not considered in this work. Finally, platforms consist of a network of computational resources that acts as a trusted base to enforce the correct assembly of on-demand system. Finally, the authors of [21] propose a solution that enables the effective transformation of medical device sensed information from the ICE framework to HL7 Fast Healthcare Interoperability Resources (FHIR) framework. This work demonstrates the feasibility of mapping the data model of the ICE architecture to profiled FHIR resources in order to achieve structural interoperability between the domains of medical devices and clinical IT infrastructure.

The previous solutions improve in different ways the current challenges of ICE systems. However, none of them considers the efficient and flexible management of the ICE components (ICE adapters, Supervisor, and communications) to ensure their security, privacy, QoS, and high availability. Additionally, this management should be made by taking into account the current environmental state (if the ICE system is under attack, emergency, or failure the actions will be different). To the best of our knowledge, this paper proposes the first solution combining NFV and SDN with MCPS to provide a flexible management of the ICE resources in real-time and on-demand.

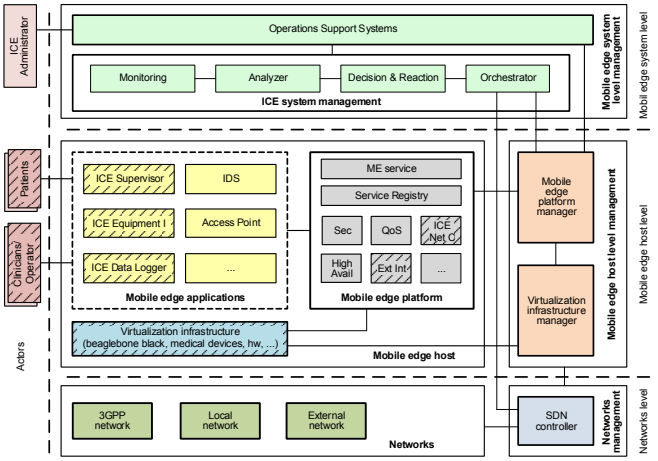

Fig. 2. ICE++ architecture oriented to the MEC paradigm

III. ICE++ Architecture

This section describes the extended ICE architecture (ICE++) that combines the MEC, SDN, and ETSI NFV [22] proposals to enable the flexible, efficient, and automatic management of the elements composing the ICE framework. Fig. 2 depicts the layers and actors of ICE++ as well as how the ICE elements are provided. These elements are depicted in boxes with striped background to show clearly how the ICE++ architecture interacts with the existing ICE framework. The internal communications between the ICE elements are not depicted to ease its understanding (they are shown in Fig. 1).

A. Mobile edge system level

Mobile edge system level management is in the upper level of our architecture and it is focused on defining and managing the behavior of the ICE components. For that, this element is composed of the *Operation Support System* (OSS) and the *ICE System Management*. The OSS deals with the logic of the ICE system. This element manages aspects such as the level of security required in emergence situations, or the QoS required in a normal state. The previous aspects are provided to the *ICE system management* to identify concrete actions and orchestrate their enforcement.

The ICE system management is composed of the next elements: *Monitoring*, *Analyzer*, *Decision & Reaction*, and *Orchestrator*. They are in charge of detecting anomaly situations and enforce the best countermeasures to address or mitigate the anomalies. Specifically, the monitoring component gathers the patients' vital signs, the network statistics, and the state of the mobile edge (ME) applications and services running in the lower level. This information is sent to the analyzer to inspect it and detect anomaly situations like, for example, congestions, attacks, infections, failures, or misconfiguration of the ICE components. Once detected, the decision & reaction component decides the required countermeasures to address

the anomaly and ensure the security, privacy, QoS, and high availability of MCPS. Among the set of potential countermeasures, we highlight the flexible and efficient deployment, configuration, relocation, and dismantlement of:

- *ME applications*. Examples of ME applications could be Intrusion Detection Systems (IDS) and Deep Packet Inspections (DPI) to detect and mitigate attacks; access points to balance the network traffic and ensure the QoS of MCPS; or ICE components (interfaces, supervisors, or applications) to guarantee the high availability of ICE.
- *ME services*. For example, a ME service could be in charge of assuring the level of security of the ICE elements. In particular, it could establish specific authentication mechanisms to ICE applications or determines a particular security protocol (e.g. TLS) between the ICE interfaces and the ICE supervisor.
- *Network infrastructure*. For example, virtual SDN-based networks to isolate ICE components according to different criteria like QoS or security. Another alternative could be the automatic management of the network communication to drop network packets belonging to a DDoS attack, or prioritize packets belonging to specific medical devices to ensure the QoS.

The Orchestrator is responsible for scheduling and triggering the previous countermeasures as well as maintaining an overall view of the ME system based on deployed ME hosts, available resources, and network topology.

B. Mobile edge host level

The mobile edge host level is focused on running ME applications, their services, and the virtualized infrastructure. This level is composed of two elements: the *Mobile edge host* and the *Mobile edge host level management*.

On one hand, the ME host provides compute, storage, and services for running ME applications. To reach it, this entity contains *Mobile edge applications*, a *Mobile edge platform*, and a *Virtualization infrastructure*. ME applications are the ICE element defined by the ICE framework as well as some applications oriented to ensure or improve the security, QoS, and high availability of the clinical scenario. Applications run as virtual machines (VM) on top of the virtualization infrastructure allocated at the edge of the network. This fact provides our solution with the flexibility, efficiency, and low latency required by ICE solutions. The virtualization infrastructure can use the hardware resources of computers, beaglebones black, or even medical devices, it depends on the scenario's configuration. ME applications interact with the ME platform to consume and provide services. Specifically, the ME platform is a set of essential services required to run ME applications on a particular virtualization infrastructure. These services can be specific for given applications or even shared for some of them. Examples of services, could be from secure communications protocols (like, for example, Transport Layer Security, TLS), to QoS protocols (like Real Time Publish Subscribe, RTPS), traffic rules control, or DNS.

On the other hand, the ME host level management is composed of two elements: the *Mobile edge platform manager* and the *Virtualization infrastructure manager*. The ME platform manager is responsible for managing the life cycle of applications including informing the ME orchestrator of relevant application related events. The Virtualization Infrastructure Manager is responsible for allocating, managing and releasing virtualized (compute, storage and networking) resources of the virtualization infrastructure, as well as collecting and reporting performance and fault information about the virtualized resources.

C. Networks level

Finally, the *Networks* level is the lowest one and it contains two elements: the *Networks* and the *Networks management*. Networks contain the physical infrastructure required to provide connectivity between the different ME applications. The Networks management contains the SDN Controller, which is able to monitor and manage in real-time and on-demand the communications of ME applications.

D. Actors level

The actors composing the proposed architecture are: *ICE Administrator*, *Clinicians & Operators*, and *Patients*. The ICE Administrator is responsible for defining the logic of the system in a high level. By using high-level statements this actor can indicate aspects like, for example, when it is required to cypher the communications of specific medical devices, what is the minimum level of QoS guaranteed by the framework, when is required to deploy a new ICE component to ensure its availability, in which states the framework has to increase the security, QoS, or high availability, etc. The clinicians & operators interact with the ME applications (ICE applications, access points, etc.), for example, to obtain the patients' vital signs or the state of the active treatments. Finally, patients, whose vital signs are sensed by the medical devices, can also interact with the ME applications to obtain medical and personal information like, for example, Electronic Health Records (EHR).

IV. Illustrative Closed-loop Clinical Challenge Scenarios

In the following subsections, we describe a challenge scenario for (i) security, (ii) QoS, and (iii) high availability. In the description of each challenge scenario we show critical concerns of current ICE solutions, and how they can be addressed by the proposed architecture.

During the last years, researchers have suggested closed-loop solutions that incorporate a pulse oximeter and infusion pump to predict life critical situations [23]. Nowadays, it is not a closed problem and important efforts are still required. While solving this problem entirely is beyond the scope of this work, we note that even when a suitable solution is realized, it will still face the concerns addressed in this paper.

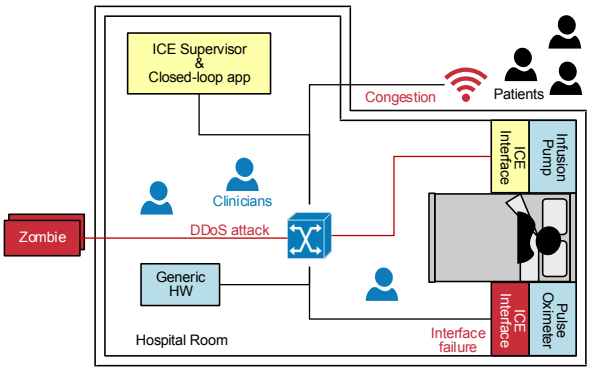

Fig. 3. Challenge scenario in the current ICE framework

Fig. 3 shows the closed-loop scenario considered in this section, which is well-known in the MCPS vision. The scenario is composed of a hospital room equipped with a patient-controlled analgesia (PCA) infusion pump and a pulse oximeter. These two medical devices do not have interoperability capabilities. Therefore, they are attached to external devices allocated in the room (raspberry Pi, beaglebone black, computer, or hardware), where ICE Equipment Interfaces run and provide interoperability. Patients' blood oxygenation (SpO2) is measured by the pulse oximeter and is sent to the ICE Supervisor through the network infrastructure. The supervisor hosts a closed-loop application in charge of receiving the SpO2, analyzing it, and communicating with the (PCA) infusion pump to start or stop delivering opioids for pain management when the percentage of SpO2 is higher or lower than a given threshold. Finally, there is an access point in the room area to allow clinicians and patients to get access to the Internet.

A. Security Scenario: Botnet DDoS attack to the ICE Components

Botnets have been recognized by Joseph Demarest, director of the FBI cyber area, as one of the most powerful threats on Internet. He stated in [24] that "each second, 18 devices are recruited by botnets" (around 567 million of devices compromised annually) estimating losses around 110,000 million dollars in the worldwide. A botnet is a network of thousands or millions of compromised devices known as bots or zombies, infected by a malware to be remotely controlled by a Command and Control (C&C) server [25]. A Distributed Denial of Service (DDoS) attack conducted by a botnet consists of a large number of zombies sending simultaneous requests to a given destiny at the same time. The detection and protection of this kind of attack is an open challenge in any network based system due to the nature of the attack and well-known security protocols that authenticate different parties or even cypher the communication are not effective.

In this context, current ICE solutions composed of medical devices with network capabilities that enable closed-loop clinical scenarios are vulnerable to botnet attacks. By considering the scenario previously defined, when the SpO2 is under a given threshold, the closed-loop application sends an alert to the infusion pump and it stops the infusion. In this context, if the infusion pump is attacked by a botnet, the pump could not continue infusing opioids, or even worse, it could not stop an active infusion, endangering the patient's life.

As it has been explained in Section II, current ICE solutions do not have capabilities to detect and mitigate botnet attacks, which supposes a critical concern. The detection of this kind of attacks requires an efficient mechanism able to analyze the network packages and flows to detect the C&C channels or the flooding packets. Additionally, once the attack has been detected, an automatic mechanism is also required to manage the network communications on-demand. This mechanism is in charge of stopping the attack by dropping the attack packets. To reach both detection and reaction capabilities, an architecture like the proposed in this paper is mandatory, which is able to manage the ICE and network resources (medical devices, medical applications, intrusion detection systems, switches, etc.) in an efficient and flexible way.

B. QoS Scenario: Overload of ICE due to emergences

Emergence situations with a high number of patients and clinicians in the ICE, can affect the QoS of ICE solutions by overloading their components and provoking unacceptable delays and latencies during the decision, reaction, and communication processes. In that sense, following our scenario, the delay in the communications between the pulse oximeter and the closed-loop application that decides if the PCA pump infuses or not opioids is critical. The problem is similar in the communication channel between the closed-loop application and the pump to stop or start the infusion. The QoS of these communications are partially solved by the state of the art by using external middlewares like DDS.

However, current solutions do not cover completely the QoS of the elements composing the ICE solution. In this sense, following the closed-loop scenario we realize that the QoS of the applications in charge of making decisions about the PCA infusion pump; the interfaces to provide network capabilities and control the medical devices; and even the access points and network infrastructure proving access to the clinical environment are not considered by current solutions. In this context, an automatic solution is needed to manage the computational, memory, and storage resources assigned to the elements composing the clinical environment. Additionally, this solution should be able to manage the life-cycle of virtual elements on-demand by considering the context.

C. High Availability Scenario: Failures or misconfiguration in ICE

This scenario depicts the drawbacks of current ICE solutions to ensure the availability of their services when they suffer failures. The availability of the elements composing the ICE scenario is a critical challenge due to the relevance of the managed information. In the proposed scenario, if in a given moment, one of the elements composing the scenario fails (the infusion pump, the pulse oximeter, their interfaces, the supervisor, or the medical application), the patient can

be injured. Despite the failure causes are not the scope of this scenario, among the possible alternatives, we highlight misconfiguration, misuse, failures, natural catastrophes, etc.

As it has been demonstrated in Section II, the majority of solutions based on ICE do not consider failures, or they just consider the replication of some elements belonging to the ICE framework. These alternatives are not valid. The former requires to install and configure manually the component that has failed, which is not acceptable due to the time restriction. On the other hand, the latter implies the reservation of physical resources that are not used during the most part of their lifetime. This is not an efficient approach because physical resources are limited, and other ICE components could need them to perform critical actions in certain moments.

In this context, a mechanism is required to deploy, manage, and dismantle the ICE framework components when they suffer a failure. Following the closed-loop scenario, if a given ICE Equipment Interface failures, the architecture proposed in this article is able to detect the situation, automatically deploy a new interface for the existing medical device, and dismantle the existing one to optimize the resources usage.

V. Deployment of a Prototype to Address the Challenges of Clinical Scenarios

The goal of this section is to demonstrate the usefulness of the proposed architecture, and show how it decides suitable actions focused on improving the challenges of Section IV. For that, we have designed several families of management policies able to detect concerns of clinical scenarios and overcome them throw different actions. In this sense, we have implemented a functional prototype of the *ICE systems management* module, making up the proposed architecture, to perform different experiments and show the time required to make the decisions needed to address the previous concerns.

A. Security, QoS, and high availability decisions

Our prototype decides specific and suitable actions to ensure the security, QoS, and high availability of the clinical environments according to management policies pre-defined by the ICE Administrator. As proof of concept, we have categorized policies into three different sets: *security*, *QoS*, and *high availability* policies. Each family of policies has a different goal. However, they share the same structure and are defined as combination of one or more sets of rules that dictate actions according to conditions. Thanks to the SDN and NFV techniques considered by our architecture, the actions of policies can be to deploy, control, or dismantle elements belonging to the ICE framework or the network infrastructure. These actions are decided by considering the current status of the network traffic, medical sensed data, and contextual information like the patients or clinicians locations. The policies are comprised by the following elements: the *type* of policy; the *element* (ICE or network components), whose information is currently being managed; the *metric* with which the network resources are evaluated; the *location* where the policy will be enforced; the *date* or the period of time at which the policy will be applied; and the *result* or set of actions to be carried out once the policy is applied.

1) Security policies: Security policies are oriented to ensure security and privacy of ICE solutions. Among the different actions that these policies can enforce, we highlight the deployment of applications like IDS or DPI, the deployment of security protocols (e.g., deploying the TLS protocol in real-time), or even the adaptation of authentication mechanisms to the level of security, etc. Additionally, these policies are also able to modify the flow tables of the SDN switches composing the clinical environment. In this sense, and following the closed-loop scenario defined in Section IV, next policy indicates that when the number of Http requests sent to a PCA infusion pump (*?infusionPump*) from unknown sources is more than a given threshold (*#UnkScrMax*), the SDN controller (*?sdnController*) will add a new rule in the flow table of the switch (*?switch*) connected to the infusion pump to drop these packets and mitigate the DDoS attack.

$$\begin{aligned}&\text{Type(\#Security)} \wedge \text{InfusionPump(?infusionPump)} \wedge \\ &\text{isConnected(?infusionPump,?switch)} \wedge \\ &\text{integer[reqs>= \#UnkScrMax] hasUnkHttpReq(?infusionPump)} \\ &\wedge \text{hasController(?switch,?sdnController)} \rightarrow \\ &\text{dropNetworkFlow(?sdnController,?reqs)}\end{aligned}$$

2) QoS policies: QoS policies manage the ME applications, services, and communications at the edge of the network to ensure the QoS of ICE solutions. Additionally, these policies can assign special privileges to medical devices, clinicians, applications, or communications to ensure their QoS. In the emergence scenario shown in Section IV, next policy detects the congestion of the network measuring the flows per second (*FPS*) processed by one of the switches of the network. Once the congestion has been detected, the policy deploys a new access point in the congested area (*?urgencyArea*) and balances the network traffic.

$$\begin{aligned}&\text{Type(\#QoS)} \wedge \text{Switch(?switch)} \wedge \\ &\text{integer[FPS in \#FPSRedAlert] hasFPS(?switch)} \wedge \\ &\text{hasLocation(?switch,?urgencyArea)} \rightarrow \\ &\text{deployAccessPoint(?urgencyArea)}\end{aligned}$$

3) High availability policies: This family is designed to cover the high availability challenge of MCPS. These policies guarantee the provision of ICE and network services to ensure the safety of the MCPS patients. Among the different actions that can be taken by these policies, we highlight the deployment on the edge of the network the ICE components as mobile applications running on top of Virtual Machines. Regarding the closed-loop scenario, the next policy detects when the interface of the pulse oximeter (*?pulseOximeter*) has a failure and deploys a new ICE interface for that device in the same area (*?room*). This policy also dismantles the interface with failure (*?oximeterInterface*) to optimize the resources.

Type(#Availability) ∧ PulseOximeter(?pulseOximeter) ∧
hasICEInterface(?pulseOximeter, ?oximeterInterface) ∧
hasState(?oximeterInterface, #Failure) ∧
hasLocation(?oximeterInterface,?room) →
deployICEInterfaceBasedOn(?oximeterInterface,?room) ∧
dismantleResource(?oximeterInterface)

B. Prototype implementation

In this Section we show the prototype implementation details of the *ICE systems management* module, making up the proposed architecture. The goal of this implementation is to have a functional implementation of the components in charge of making suitable decisions to ensure the security, QoS, and high availability of the clinical environment.

For that, we have implemented a prototype of the ICE management system by using Semantic Web techniques. Specifically, the previous policies are expressed as semantic rules by using SWRL (Semantic Web Rule Language). The information managed by the architecture (ICE elements, medical devices, patients, clinicians, network statistics, network services, etc.) is shaped in an ontology, which is defined in OWL 2 (Web Ontology Language). We have chosen OWL 2 rather than other languages like RDF, RDFS, or DAML+OIL because OWL 2 is more expressive than the rest. It was specifically designed as an ontology language; it is an open standard; and it is the main ontology language used nowadays in Semantic Web. The Jena API generates an ontological model with the information shaped in the ontologies and policies. To infer new knowledge, we have used Pellet as semantic reasoner, which receives the ontological model and returns an model with new knowledge.

C. Experiments

This section some experiments are performed to measure how much time is required by the *ICE systems management* module to make decisions and ensure the security, privacy, QoS, and high availability of ICE solutions. For that experiments we have used a workstation with 32GB of RAM, a six-core Intel i7-5930K at 3.5GHz with hyper-threading running Linux, and one NVIDIA GeForce GTX 1080 with 8GB RAM.

TABLE I
INDIVIDUAL DISTRIBUTION OF POPULATION

Element	Amount	Percentage
Patient beds	650	1.1%
Clinicians	1,020	2.1%
ICE elements	3,538	6.3%
Patients data	12,800	22.6%
Network resources	700	1.2%
Network data	35,600	63.2%
Others	2,024	3.5%
Total	**56,332**	**100%**

A way to measure the performance of the ICE management system is making executions with different levels of complexity. This complexity is related to the number of individuals considered in the ontology. Increasing the number of individuals will provoke an increment of the complexity of the executions. The number of individuals contained in our ontologies is referred as *population*. This was randomly generated for the experiments, but in a controlled way in order to achieve the desired distribution for simulating a scenario as real as possible. Table I depicts the number of elements used in our environment and the percentages obtained for them. The distribution of the different elements and the ratio patient beds versus clinicians have been measured by considering the information of the American Hospital Directory [26].

Another issue to evaluate the Reasoner scalability is the way in which the population sizes are established. In this sense, considering the distribution of individuals shown in Table I, we defined an initial population of 10,000 individuals and we increased this population with other 10,000 individuals in each step. Table II shows the relationships between the individuals and the statements generated by the Reasoner. As observed, the number of statements (obtained after the reasoning process) is proportionally increased according to the individuals number. Each population group is used to obtain the time required to check the knowledge base consistency and infer new information. Fig. 4 depicts this time, measured in milliseconds (ms), used by the Reasoner to validate the ontology.

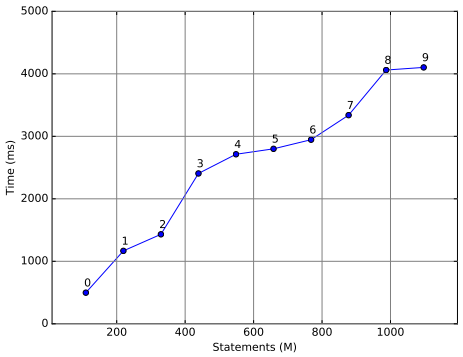

Fig. 4. Decision time

Comparing the increase of individuals and statements with the reasoning time, we can observe that the proposed architecture can support a very large number of individuals within a reasonable reasoning time. Furthermore, the linearity property behind these results allows us to deduce that a better computer system setting would obtain lower reasoning times. As conclusion of this section, we have demonstrated that the proposed architecture is able to make decisions to ensure the security, privacy, QoS, and high availability of MCPS in an acceptable time. In this sense, for a hospital with 650 beds, which is more or less the capacity of the Einstein Medical Center Philadelphia (one of the biggest of the Pennsylvania state), our solution needs slightly less than 3,000 ms to detect an anomaly situation and decide the reaction.

TABLE II
NUMBER OF INDIVIDUALS AND STATEMENTS PER POPULATION

Population	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
Individuals	10,000	20,000	30,000	40,000	50,000	60,000	70,000	80,000	90,000	100,000
Statements	109,852	219,532	329,212	438,892	548,572	658,252	767,932	877,612	987,292	1,096,972

VI. CONCLUSION AND FUTURE WORK

This paper presents a novel architecture to manage the security, privacy, QoS, and high availability of solutions based on Integrated Clinical Environments (ICE). The proposed architecture is oriented to the MEC paradigm, which manages resources allocated in the edge of the network, providing low latency required to ensure QoS; NFV techniques enable the high availability, scalability and efficiency required by ICE solutions; and the SDN paradigm guarantees the communication management of the ICE components in real-time and on-demand. Additionally, three scenarios depict the drawbacks of current ICE solutions and how the proposed architecture deals with them. Finally, some experiments show the performance of our architecture making decisions to ensure the challenges of MCPS.

As next step, we are implementing and validating the ME host and network levels of the proposed architecture to show the time required to deploy, configure, and dismantle ICE components and network applications. For that end, we are using OpenStack as VIM to deploy and instantiate VM and networks where the ICE Supervisor and Interfaces are running; OpenDaylight as SDN Controller to control both physical and virtual switches composing the network topology; and Open Baton to orchestrate the SDN and NFV planes.

ACKNOWLEDGMENT

This work is supported by a Séneca Foundation grant within the Human Resources Researching Postdoctoral Program 2017 and a Government of Ireland Postdoctoral fellowship 2018.

REFERENCES

[1] J. A. Stankovic, "Research Directions for Cyber Physical Systems in Wireless and Mobile Healthcare," ACM Transactions on Cyber-Physical Systems, vol. 1, no. 1, pp. 1:1–1:12, 2017.

[2] D. Arney, J. Plourde, and R. Schrenker, P. Mattegunta, S. F. Whitehead, and J. M. Goldman, "Design Pillars for Medical Cyber-Physical System Middleware," Proceedings of the 5th Workshop on Medical Cyber-Physical Systems, pp. 124–132, 2014.

[3] ASTM International, F2761-09, "Medical Devices and Medical Systems - Essential safety requirements for equipment comprising the patient-centric integrated clinical environment (ICE) Part 1: General requirements and conceptual model," 2013.

[4] K. K. Scott, F. Benjamin, J. Taylor, M. D. Kyle, "Cybersecurity in healthcare: A systematic review of modern threats and trends," Technology and Health Care, vol. 25, no. 1, pp. 1–10, 2017.

[5] S. G. Langer, "Cyber-Security Issues in Healthcare Information Technology," Journal of Digital Imaging, vol. 30, no. 1, pp. 117–125, 2017.

[6] T. Shah, A. Yavari, K. Mitra, S. Saguna, P. P. Jayaraman, F. Rabhi, R. Ranjan, "Remote health care cyber-physical system: quality of service (QoS) challenges and opportunities," in IET Cyber-Physical Systems: Theory & Applications, vol. 1, no. 1, pp. 40–48, 2016.

[7] Food and Drug Association (FDA), Medical Device Reporting (MDR), https://www.fda.gov/MedicalDevices/Safety/default.htm, Accessed on February 2018

[8] A. Ahmed and E. Ahmed, "A Survey on Mobile Edge Computing," Proceedings of the 10th International Conference on Intelligent Systems and Control, pp. 1–8, 2016.

[9] J. Matias, J. Garay, N. Toledo, J. Unzilla, and E. Jacob, "Toward an SDN-Enabled NFV Architecture," IEEE Communications Magazine, vol. 53, no. 4, pp. 187–193, 2015.

[10] L. Gu, D. Zeng, S. Guo, A. Barnawi, and Y. Xiang, "Cost Efficient Resource Management in Fog Computing Supported Medical Cyber-Physical System," IEEE Transactions on Emerging Topics in Computing, vol. 5, no. 1, pp. 108–119, 2015.

[11] D. Arney, J. Plourde, and J. Goldman, "OpenICE medical device interoperability platform overview and requirement analysis," Biomedical Engineering/Biomedizinische Technik, In press, 2017

[12] D. Norris, "The Internet of Things: Do-It-Yourself at Home Projects for Arduino, Raspberry Pi and BeagleBone Black", Tab Electronics, 2015

[13] O. Koksal, and B. Tekinerdogan, "Obstacles in Data Distribution Service Middleware: A Systematic Review," In Future Generation Computer Systems, vol. 68, pp. 191–210, 2017.

[14] S. Hamed, D. Arney, and J. Goldman. "Toward a Safe and Secure Medical Internet of Things," 2016.

[15] H. Nguyen, B. Acharya, R. Ivanov, A. Haeberlen, and L. T. X. Phan, O. Sokolsky, J. Walker, J. Weimer, W. Hanson and I. Lee, "Cloud-Based Secure Logger for Medical Devices," Proceedings of the IEEE First International Conference on Connected Health: Applications, Systems and Engineering Technologies, Washington DC, pp. 89–94, 2016.

[16] L. Cheng, Z. Li, Y. Zhang, Y. Zhang, and I. Lee. "Protecting interoperable clinical environment with authentication," SIGBED Review, pp. 34–43, 2017.

[17] V. P. Ranganath, Y. J. Kim, J. Hatcliff and Robby, "Communication patterns for interconnecting and composing medical systems," Proceeding of the 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, Milan, pp. 1711–1716, 2015.

[18] M. Garca-Valls, I. E. Touahria, "On Line Service Composition in the Integrated Clinical Environment for eHealth and Medical Systems," Sensors, vol. 17, no. 6, 2017.

[19] B. Almadani, B. Saeed, and A. Alroubaiy, "Healthcare systems integration using Real Time Publish Subscribe (RTPS) middleware," In Computers & Electrical Engineering, vol. 50, pp. 67–78, 2016.

[20] W. Zhao, and M.Q. Yang,"Dependability enhancing mechanisms for integrated clinical environments," The Journal of Supercomputing, vol. 73, no. 10, pp. 4207–4220, 2017.

[21] B. Andersen, M. Kasparick, H. Ulrich, S. Schlichting, F. Golatowski, D. Timmermann, and J. Ingenerf, "Point-of-care medical devices and systems interoperability: A mapping of ICE and FHIR," Proceedings of the IEEE Conference on Standards for Communications and Networking, Berlin, pp. 1–5, 2016.

[22] ETSI NFV ISG, "Network Functions Virtualisation (NFV); Network Operator Perspectives on NFV priorities for 5G," Technical report, 2017.

[23] M. Pajic, R. Mangharam, O. Sokolsky, D. Arney, J. Goldman, and I. Lee, "Model-Driven Safety Analysis of Closed-Loop Medical Systems," IEEE Transactions on Industrial Informatics, vol. 10, no 1, pp. 3–16, 2014.

[24] J. Demarest, "Taking down botnets: Public and private efforts to disrupt and dismantle cybercriminal networks (Statement Before the Senate Judiciary Committee, Subcommittee on Crime and Terrorism)", http://www.fbi.gov/news/testimony/taking-down-botnets, 2014.

[25] K. Alieyan, A. ALmomani, A. Manasrah, and M. M. Kadhum, "A survey of botnet detection based on DNS," Neural Computing and Applications, vol. 28, no. 7, pp. 1541–1558, 2017.

[26] American Hospital Directory, https://www.ahd.com/states/hospital_PA.html, Accessed on February of 2018.