CareNet: Building a Secure Software-defined Infrastructure for Home-based Healthcare

Peilong Li Chen Xu Yan Luo
Department of Electrical and Computer Engineering
University of Massachusetts Lowell

Yu Cao
Department of Computer Science
University of Massachusetts Lowell

Jomol Mathew
Department of Information Technologies
University of Massachusetts Medical School

Yunsheng Ma
Department of Medicine
University of Massachusetts Medical School

ABSTRACT

Healthcare network and computing infrastructure is rapidly changing from closed environments to open environments that incorporate new devices and new application scenarios. Home-based healthcare is such an example of leveraging pervasive sensors and analyzing sensor data (often in real-time) to guide therapy or intervene. In this paper, we address the challenges in regulatory compliance when designing and deploying healthcare applications on a heterogeneous cloud environment. We propose CareNet framework, consisting of a set of abstraction and APIs, to allow the specification of compliance requirements. This work is a collaboration among computer scientists, medical researchers, healthcare IT and healthcare providers, and its goal is to reduce the gap between the availability of software defined infrastructure and meeting regulatory compliance in healthcare applications.

Keywords

Software Defined Infrastructure; HIPAA Compliance; Home-based Healthcare; CORD; API

1. INTRODUCTION

The advances in information technology greatly accelerates the innovations in healthcare technology recently. In particular, home-based healthcare services such as rehabilitation, telemedicine, and so on are being realized due to the availability of low-cost sensors, effective data processing capabilities and advanced networking technologies. The prediction reveals the demands of home-based healthcare market will keep increasing rapidly by at least 5% per year to the year of 2020, thanks to the cost savings and the comfort provided to patients and people in need.

The growing trend of home-based healthcare has introduced new challenges in data collection, transfer and sharing since the patients and their care providers are often geographically distributed. Existing healthcare infrastructures such as the traditional close-environment healthcare and the emerging cloud-based healthcare face vital obstacles as follows. Firstly, traditional healthcare assumes a closed environment in a single or multiple fixed location, which cannot efficiently analyze and share the patient data securely to multiple stockholders. For example, the sensor data collected on patient from her residence have to be transferred to a remote analytics service or to the doctors' offices for diagnosis. Such data transfer over public networks requires both intensive computing resources and sufficient protection which are new concepts to traditional healthcare. Secondly, the reliability of cloud-based healthcare hinges on the data transmission performance between the end devices and the cloud. Many emerging home-based mission critical healthcare services that require real-time responses and decisions demand the network to be low-latency and high-bandwidth. However, real-world cloud latency ranges from hundreds of milliseconds to a few seconds because of the structure of the Internet. Therefore, cloud by itself is not a feasible solution to home-based healthcare. Thirdly, all patient data related diagnosis and analytics activities should be supported with an infrastructure that is regulation compliant. Patient information must be protected to comply with regulations such as Health Insurance Portability and Accountability Act (HIPAA).

The emerging software-defined infrastructure (SDI) has shed light on the challenges in existing healthcare infrastructure. While computing resources on cloud platforms are flexible and cost-effective, there are new resources provisioned at the network edges for applications requiring high throughput and low latency. CORD [1] is such an example of edge based computing platform residing in the central office of a telco. Paradrop [2] contains programmable resources in a WiFi router deployed inside a patient's premises. These heterogeneous resources at every part of the network (end point, edge and core) bring both unprecedented opportunities for application design and challenges of performance and compliance verification.

There exists a gap between the availability of emerging SDI and deploying regulation compliant healthcare services on top of that. In this paper, we are motivated to achieve the following goals. (1) We propose a healthcare framework called "CareNet" that enables the employment of a heterogeneous home-edge-core cloud to render high performance

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

SDN-NFV Sec'17, March 22-24, 2017, Scottsdale, AZ, USA
© 2017 ACM. ISBN 978-1-4503-4908-6/17/03...$15.00
DOI: http://dx.doi.org/10.1145/3040992.3041007

and real-time responsiveness for the home-based healthcare services. (2) We propose a secure end-to-end data transmission mechanism and an advanced access control scheme so that every networking transactions on CareNet has to be compliant with the HIPAA technical safeguard. (3) We design a suite of high level Application Programming Interfaces (API) that exploit the underlying SDI resources to help various stockholders to express their workflow and simplify the management of the healthcare resources without knowing the technical details. This work is intended to initiate the discussion among network researchers, medical researchers, healthcare ITs, patients and clinical staffs.

This paper is organized as follows. Section 2 provides the background of the work. Section 3 and Section 4 present the design of CareNet system and the design of CareNet APIs respectively. Finally, we discuss the limitations of the work and future directions in Section 5.

2. BACKGROUND

2.1 Software Defined Infrastructure for Healthcare

Recently, the technology to exploit abundant computing and networking resources at telco central offices on the network edge, has enabled the feasible development of an interactive home-based healthcare model for care providers. As depicted in Figure 1, a home-based healthcare model can consist of three major components: 1) **edge** - at homes, patients' sensor data are collected by various monitoring devices and the sensor data are transmitted and aggregated on a computing device (called "HomeNode") such as an enhanced WiFi router at the premise. At the network edge, telcos provide computing racks and white box switches [1] to support flexible data processing right after the data streams leave patients' homes. It is vital to keep computing resource at the edge of the network to support latency-sensitive applications and services. The edge nodes can also support intensive computation and stream mining, which process the data and reduce the data volume at a very early stage, thus cutting down delays and saving network bandwidth; 2) **hospital private cloud** - only hospital ITs and doctors can access private cloud. The private cloud serves as an enterprise scale data center and can be used to store and process patient medical records. But the private cloud is optional for some care providers such as rural clinics; 3) **public cloud** - both patients and doctors can access the public cloud. The public cloud provides extra richer computation and storage for data analytics, and hosts REST service for mobile and web applications.

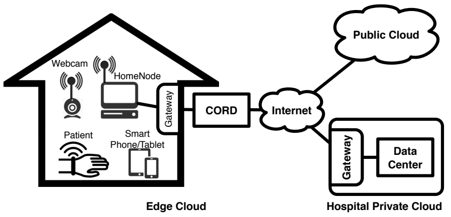


Figure 1: Home-based Healthcare

The techniques of SDI is fundamental of the proposed home-based infrastructure, thanks to the programmability of network with software-defined networking (SDN) and the feasibility of resource management in cloud with OpenStack. Recently, the novel Everything-as-a-Service Operating System (XOS) emerges to provide an interface so that application builders can easily leverage the underlying programmable infrastructure. XOS defines a coherent framework that consists of both OpenStack and OpenNetwork Operation System (ONOS) for combining SDN, network function virtualization, and cloud services, all running on commodity hardware, to build a cost effective and agile cloud infrastructure.

2.2 Regulatory Compliance Requirements

There are three major components in complying with the US HIPAA standards: Administrative, Physical, and Technical. These guidelines stipulate that all medical practices must ensure that all necessary measures are in place while saving, accessing and sharing any electronic medical data to keep patient data secure. Lack of compliance to the HIPAA security standards could lead to large fines and in extreme cases even loss of medical licenses. While the administrative and physical safeguard guidelines pertain mostly for employees' security awareness and training, and facility related access control and security, the technical safeguard regulates the data storage and retrieval and the security of the network, which is addressable with computer techniques. We therefore only focus on the technical safeguard in this paper.

Technical safeguards are becoming increasingly important due to technology advancements in healthcare. Care providers are faced with the challenge of protecting electronic protected health information (ePHI), such as electronic health records, from various internal and external risks. To reduce risks to ePHI, covered entities must implement technical safeguards as good business practices. Table 1 lists a collection of Technical Safeguard standards and certain implementation specifications, which includes 4 regulation sections: access control, integrity, person/entity authentication, and transmission security. A covered entity may use appropriate security measures that allow it to reasonably implement the standards.

Table 1: Technical Safeguard Requirements

Standards	Sections	Explanation
Access Control	§164.312(a)(1)	Unique User Identification (avoid disclosure of user information)
		Emergency Access Procedure: procedures for obtaining necessary ePHI during an emergency (privilege endorsement)
		Encryption and Decryption: a mechanism to encrypt and decrypt ePHI
Integrity	§164.312(c)(1)	Mechanism to Authenticate ePHI
Person/Entity Authentication	§164.312(d)	Implement procedures to verify that a person or entity seeking access to ePHI is the one claimed
Transmission Security	§164.312(e)(1)	Integrity Controls: the security measures to ensure that electronically transmitted ePHI is not improperly modified without detection until disposed of
		Encryption: a mechanism to encrypt ePHI whenever deemed appropriate.

3. SYSTEM DESIGN

To leverage the emerging SDI technologies in a regulation compliant manner, we propose in this section the CareNet, a heterogeneous computing and networking framework for providing effective healthcare to people living in a home setting equipped with advanced and versatile sensors. The goal of this section is to present the high level architecture of CareNet, and describe the key components in this framework. We also outline a comprehensive patient data processing/accessing/sharing mechanism that is part of the CareNet framework to enforce the regulation compliance.

3.1 The CareNet Framework

A high-level overview of CareNet framework is shown in Figure 2. The major distinctions between the conventional healthcare network and CareNet are the deployment of sensors at patients' premises and the presence of heterogeneous edge/cloud computing resources at different segments of the system.

We explain the architecture of the system with the running flow of sensor data as follows. Firstly, in this human-centric framework, the healthcare activities are driven by sensor data generated around patients. Abundant body sensors and monitoring devices are installed on patient's body or at patient's home. Then, the heterogeneous sensor data stream is aggregated and preprocessed at an enhanced WiFi router or a small compute system called "HomeNode". The HomeNode runs a daemon process to associate each data stream from the sensors to an isolated application in a docker container for processing. Secondly, the container applications communicate with the CORD edge cloud at telco's central office that is equipped with rich software-defined compute, storage, and network resources. Leveraging such edge computing resources greatly reduces raw data volume that needs to be transferred, and highly reduces the response latency for some time-sensitive applications. The CareNet API, as a higher level abstraction of XOS running on CORD, renders the interface to manage the underlying edge resources. Every CareNet API call must be authenticated with the methods described in the next subsection to ensure regulatory compliance. Thirdly, the preprocessed data will be encrypted and then reach to the hybrid cloud domain via the Internet for more powerful and scalable computing and storage. The hybrid cloud also hosts RESTful service for all parties to access the information from their web or mobile applications.

3.2 Regulation Compliance

As regulatory compliance and heterogeneous computing resources are introduced into the proposed infrastructure, we have to consider more emerging issues on data security as follows.

Firstly, how to design a secure end-to-end network framework that consists of all the CareNet components - the HomeNode, the edge cloud and the core cloud, as regulated by HIPAA transmission security (164.312 (e))? We propose the secured networking architecture in Figure 3. From left to right, the solid arrows represent the data flow. Security-sensitive network flows that come from HomeNode will pass through an Internet Protocol Security (IPsec) protected network link to avoid wiretapping. Once data is processed after arriving at ISP's CORD cloud, it will be encrypted before going into the untrusted network domain. For clients to access the processed data, they need to acquire a proper security key managed by the authentication system that comes with the CareNet API. It is worth noticing that data only flows from trusted domain to untrusted domain in one direction so that patient information won't be tampered by malicious information. From right to left on Figure 3, the dotted arrows elaborate the management flow. Commands and requests from clients' side have to pass through a double authentication scheme before entering the trusted domain. First of all, to establish the connection between clients and CareNet server at CORD over the Transport Layer Security (TSL) tunnel, clients need to acquire a proper certificate such as a SSH certificate. Second of all, requests made by calling the CareNet API will need to be authenticated. The authentication process will be elaborated in detail in the next point. The double authentication scheme ensures both network connection and access control are secure.

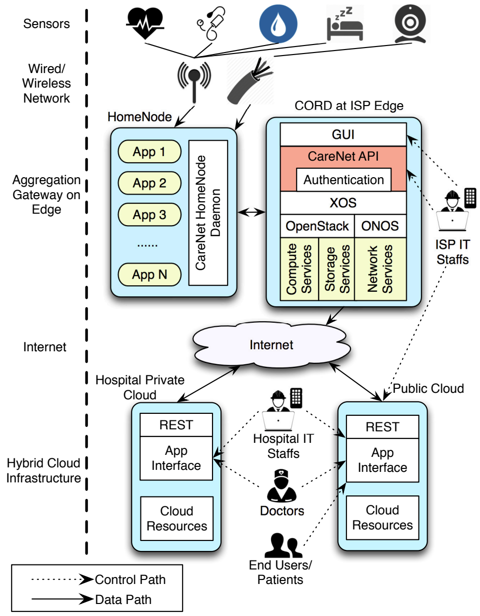


Figure 2: Architecture of CareNet

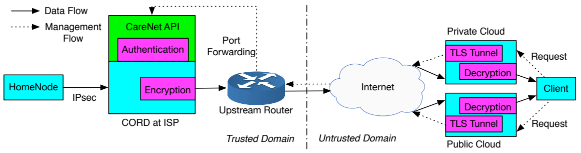


Figure 3: The Transmission Security Compliant Network Framework

Secondly, HIPAA regulates another three important standards in technical safeguard: access control (164.312 (a)), integrity (164.312 (c)), and person or entity authentication (164.312 (d)), which require the integrity and protection of ePHI under the agreement with multiple parties. Specifically, patients have the right to delegate permissions to different data consumers and the permissions are subject to change in different situations such as time-out and security key revocation. Therefore, based on the relationship between the data owner and data consumers, we divide the authentication system into public/private domains, where in private domain (parents, relatives, friends) we apply role-based access control [3] and in public domain (doctors, researchers) we use attribute-based access control [4].

4. THE CARENET API

To facilitate the usability of CareNet framework, we need well-defined and high-level APIs whereby both technical people like network operators and application developers, and non-technical people such as care providers and patients can express their requirements on data collection, sharing and processing. In this section, we first introduce the abstraction model in CareNet framework, and then present the APIs and explain their usage in details.

4.1 CareNet Abstraction

The high-level abstraction of CareNet framework aims to explain the major roles of objects and how they may interact with each other to express the workflow. The overall abstraction include Patients, Services, Groups, Resources, Users, Policy, and Policy Repository, which is illustrated in Figure 4. Because of page limitation, we explain the meaning of each component on the API introduction page [5].

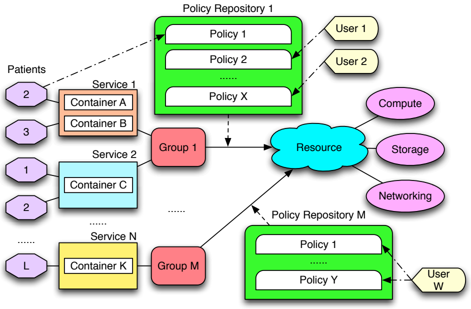


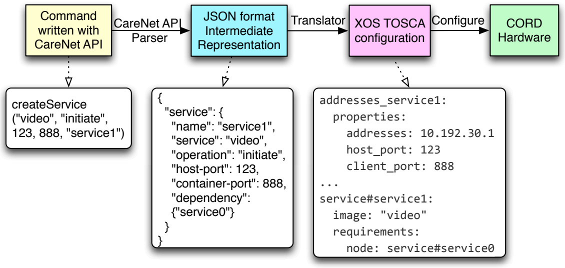
Figure 4: API Abstraction

4.2 CareNet APIs

We propose a list of APIs to facilitate the management of CareNet with the aforementioned abstractions to serve the purpose of 1) service management, 2) performance management, and 3) policy enforcement. The detailed API definition and description are depicted on the CareNet API introduction page [5] and we demonstrate a concrete use case by leveraging the APIs in the use case study page [6]. The design of the APIs renders a unified interface for care providers to manage the proposed CareNet framework and promotes the flexibility of policy specification and compliance enforcement.

4.3 CORD Configuration with CareNet APIs

We design an automatic CareNet API conversion mechanism to help translate commands written with CareNet APIs into CORD hardware configurations. As demonstrated in Figure 5, the CareNet system allows users to first specify their requirements through the CareNet APIs or use web/app graphic user interfaces that are built upon the CareNet APIs. We call this step the requirement submission. Then the user requirement submission that are written with CareNet API will be fed into our designed API parser. The API parser applies regular expression (RegEx) technique to extract the keywords from requirement submission and find the argument domains within each API function to generate a JSON-formatted intermediate representation (IR). After obtaining the JSON IR, we use a translator to map the key-value pairs in IR to a Topology and Orchestration Specification for Cloud Applications (TOSCA) [7] formatted configuration file. Since TOSCA file is used as the interface configuration to the XOS system, our mapped TOSCA configuration can then eventually configure the CORD hardware.



```
createService
("video", "initiate",
123, 888, "service1")
```

```
{
  "service": {
    "name": "service1",
    "service": "video",
    "operation": "initiate",
    "host-port": 123,
    "container-port": 888,
    "dependency":
    {"service0"}
  }
}
```

```
addresses_service1:
  properties:
    addresses: 10.192.30.1
    host_port: 123
    client_port: 888
...
service#service1:
  image: "video"
  requirements:
    node: service#service0
```

Figure 5: Configure CORD Hardware with CareNet APIs

5. DISCUSSION

We would like to point out some limitations of this work and a few future directions. First, the proposed abstraction and APIs are not a comprehensive coverage of the regulatory requirements. As the healthcare related laws are complex, it is extremely difficult to express all requirements of a healthcare application especially when persons such as healthcare providers are not proficient with technologies in computing and networking. There should be a close collaboration among networking researchers, medical researchers and healthcare and clinic personnel, who together can refine the APIs and learn from using the APIs in real clinical settings. Second, the mapping of the APIs to underlying SDI is still an undergoing work. For now we are able to identify some critical API functions that are sophisticated enough to express the workflow in some known use cases. In the near future, we plan to explore a more modularized compiler design for the CareNet language.

6. REFERENCES

[1] L. Peterson, "Cord: Central office re-architected as a datacenter," Nov. 2015. [Online]. Available: http://sdn.ieee.org/newsletter/november-2015/cord-central-office-re-architected-as-a-datacenter

[2] "Paradrop official website," 2015. [Online]. Available: https://www.paradrop.io/

[3] R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman, "Role-based access control models," *Computer*, vol. 29, no. 2, pp. 38–47, Feb 1996.

[4] S. Yu, C. Wang, K. Ren, and W. Lou, "Achieving secure, scalable, and fine-grained data access control in cloud computing," in *2010 Proceedings IEEE INFOCOM*, March 2010, pp. 1–9.

[5] P. Li, "The carenet api introduction," 2016. [Online]. Available: https://github.com/ACANETS/CareNet/blob/master/API.md

[6] ——, "A case study with carenet," 2016. [Online]. Available: https://github.com/ACANETS/CareNet/blob/master/Case.md

[7] O. A. open standards for the information society), "Tosca simple profile in yaml version 1.0," 2016. [Online]. Available: http://docs.oasis-open.org/tosca/TOSCA-Simple-Profile-YAML/v1.0/cs01/TOSCA-Simple-Profile-YAML-v1.0-cs01.pdf