A Framework for Secured Collaboration in mHealth

Logan Widick, Josh Talkington, Garima Bajwa and Ram Dantu
Department of Computer Science and Engineering
University of North Texas, Denton, Texas 76201
(loganwidick, joshtalkington, garimabajwa)@my.unt.edu
rdantu@unt.edu

***Abstract*—We have designed a novel framework of services, protocols and technologies to ensure the secure collaboration in M2M networks, specifically in mobile health. The promise of mobile health to reform preventive self-care opens new doors for remote monitoring to improve health care communication. With cardiopulmonary resuscitation (CPR) as an example, we classify our M2M elements into services, roles, human-computer protocols and technologies through which we require trust, anonymity, scalability, and active detachment. We simulate a scenario in which a patient needs CPR and through the use of widely available technologies (such as a smartphone and secure web sockets) we demonstrate a technological collaboration that facilitates secure emergency mobile health services.**

***Keywords*—Cardiopulmonary resuscitation(CPR); mobile health; machine to machine(M2M); secure collaboration**

I. INTRODUCTION AND BACKGROUND

With the rise of ever increasing heterogeneous networks, technological collaboration is an inevitable point in the aim to tackle real-world problems. Resources need to be shared and effectively combined to embody the dynamic and responsive spirit of our next generation approaches. With any new effort comes an evolution of security paradigms. Security mirrors the very nature of what it is securing, and when we talk about collaboration of different elements, whether it be humans, services or computer protocols, we must ensure that the fundamental features are preserved.

Collaborative networks are growing into a new scientific discipline as a result of advances in communication and information technology, and the market and societal needs [1]. These collaborations have penetrated every domain, with applications ranging from social media, business models to cyber security, smart energy, and healthcare [2]–[5]. In order to fuel the adoption of advancing technology collaborations, there is a greater demand for secure collaboration. Machine-to-Machine (M2M) collaborations alone cannot guarantee all the solutions [6]. There is a lack of consensus on the multitude of technologies that could be used for M2M [7] [8].

Mobile health (mHealth) is the next natural step in health care. It provides us with a good case study of a collaborative process, especially when you consider emergency health care. In an emergency situation, a demand for fast, robust, and secure protocols are essential. In order to effectively meet these demands, any available technology must have the ability to help. MHealth is also a unique M2M application. Many of the outcomes and interactions in healthcare are based on complex human protocols. For example, a patient may want a second opinion, or a doctor may need to perform an action based on decades of experience. When these protocols are supplemented in a M2M fashion, additional requirements are needed. M2M collaborative implementations need to be effectively paired with secured Human/Computer protocols through communication tools to allow interventions that cannot be simply managed by M2M collaborations. This issue becomes even more pressing in mHealth scenarios where the users of the system need to explore the data seamlessly and collaboratively for effective local and remote provisioning of health services [9], [10].

Smartphones and other mobile devices are capable of collecting and sending lifesaving data. Mintz-Habib et al. [11] proposed the original prototype for emergency services in VoIP environments. Chandrasekaran et al. [12] designed and implemented a remote media control (RMC) system based on the Session Initiation Protocol (SIP) widely used in Voice over Internet Protocol (VoIP) applications. RMC allowed emergency dispatchers to control a victims or bystanders camera and microphone to remotely view what is happening in an emergency situation. However, the existing RMC implementation involved communication between a single victim or bystander and a single dispatcher. In our proposed framework we highlight an interoperable secure collaboration platform among these different technologies along with Human/Computer protocols to support mHealth services in multiple ways without having to worry about the underlying technologies of communication.

II. REQUIREMENTS

For any scenario, threats can come from a variety of directions, motives, and domains. They aim to disrupt the outcomes of a system through attacks. Standard security attributes such as confidentiality, integrity, and availability are well known, but M2M brings with it new challenges and priorities. In a setting of collaboration, different elements have different priorities than traditional networked solutions. This calls for additional security features to be respected. For a secure, collaborative, M2M protocol, we propose four additional requirements; trust, anonymity, scalability and active detachment.

978-1-4673-7648-8/15/$31.00 ©2015 IEEE

Trust emphasizes validity and ensures quality. All the entities need to trust each other, as to not spend resources on verification. It is also important for the trust to be transitive.

Anonymity alleviates privacy concerns. Anonymity is essential to provide unbiased, fluid response and also promotes scalability.

Scalability refers to the dynamic distribution of protocol functions during collaboration to multiple elements. This would include scaling up to many parties and technologies.

Active detachability ensures that the entities involved can independently exercise their duties in case of abnormal loss of communication ability. This also requires the elements to be persistent in restoring communication.

These requirements define what it takes to protect an M2M system from harm. Given each of these, we aim to uphold the success of an applied situation to our framework.

III. FRAMEWORK

Every effective collaborative process has essential parts and requirements. M2M is unique in that it brings special interactions that have not been considered in a secure way. Figure 1 shows an overview of our framework. Our framework consists of four layers: services, roles, human/computer protocols, and underlying technologies.

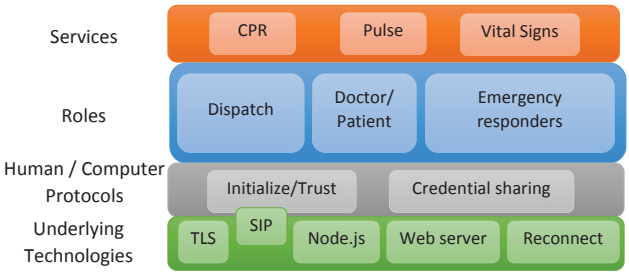

Figure 1. The layers of the framework for secure collaboration

The Services layer describes what tasks could be performed. Services provide needs with outcomes. Services also differ from other services in that their needs, along with their outcomes, change. An example of a service could be a smart meter. A smart meter fulfills a need to streamline meter management by producing the outcome of usable meter analytics. This would differ from a logistics service, where the needs and outcomes are different. In order to fulfill the needs and produce outcomes, services utilize different elements. When these elements become a part of our service we need to organize them. We do that through our next layer, the Roles layer.

Roles are service elements that have different functions. Each role would facilitate different interactions needed to ultimately carry out the service. For a video surveillance service, the roles might include a camera, manager, and network. Whatever the M2M service, we can classify different roles into workers, helpers and intermediaries. Workers provide the need and epicenter for collaboration. For example, a person administering CPR would be a worker. Helpers help, organize and conclude. An example of a helper would be a physician that is providing feedback concerning the effectiveness of CPR administration. Our definition of an intermediate means any technology that facilitates a worker/helper relationship, such as a server or a network.

Depending on the service, roles interact in many ways. In a M2M digital billboard service, a digital billboard would be a worker, and a content server would be a helper. During CPR administration, the workers are the people administering CPR, while the helpers can be doctors that are providing feedback concerning the effectiveness of CPR administration. The roles in the digital billboard service do not have the same interactions as the roles in CPR administration. We strive to preserve these interactions through a third layer, the Human/Computer protocols layer. This layer describes the specific protocols that govern interaction between worker and helper devices and the people that control them. This layer is the most diverse and telling. Each service brings with it a new set of interactions that need to be respected and maintained, especially in mHealth. For example, in the context of CPR administration, the relationships and interactions between bystanders and doctors are included in the Human-Computer Protocols layer.

A diverse set of protocols and interactions calls for a wide range of technologies. The Underlying Technologies layer describes the technologies that enable the interactions described in the Human/Computer Protocols layer. Each technology is there to support a requirement of any element above it in the framework. For example, it would be common to include TLS for encryption, but encryption might not be necessary for all applications.

IV. MEETING THE REQUIREMENTS

Using these four layers (services, roles, human/computer protocols, and underlying technologies), we provide easier avenues to obtain our requirements. Consider a scenario in which worker, helper, and intermediate roles were presented together as one monolithic module instead of as separate modular components. Assume that there is one entity A with this monolithic module and is trying to perform a task. If entity A requested guidance, then another entity B with the same monolithic module would need to be added. However, entity B does not need the worker and intermediate portions of the monolithic module. If either entity A or entity B requested additional guidance, another entity C with the same monolithic module would need to be added. Again, this leads to redundancy as entity C does not need the worker and intermediate portions of the monolithic module. The redundancy caused by the addition of entities B and C that have the monolithic module instead of helper-specific modules will eventually decrease scalability. Therefore, by contradiction, the separation of worker, helper, and

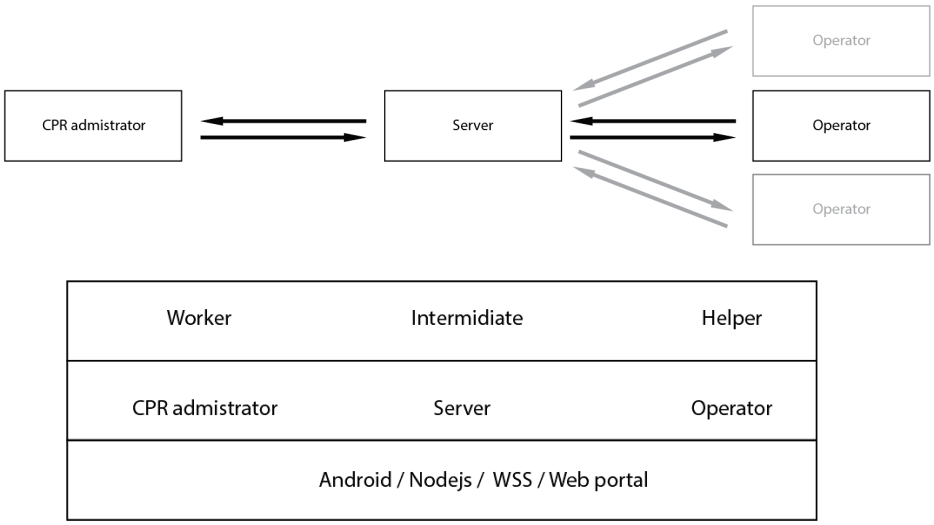

Figure 2. Representation of the relationships between elements of our framework for secure collaboration in the CPR case study

intermediate roles helps the framework scale to 1 worker and many helpers.

Again, assume that the worker, helper, and intermediate roles were presented together as one monolithic module instead of as separate modular components. Assume that entity D is performing a task and is receiving guidance from another entity E. At a point in time T1, additional work needs to be done, and entity D cannot perform this additional work. Thus, entity D will require assistance from another entity F, which also has the monolithic module. However, entity F does not need the helper and intermediate portions of the monolithic module. At a later point in time T2, additional work comes in, and another entity G will need to be added to do this work. Again, entity G comes with the monolithic module but does not need the helper and intermediate portions. The redundancy caused by the addition of entities F and G that have the monolithic module instead of worker-specific modules will decrease scalability. Therefore, by contradiction, the separation of worker, helper, and intermediate roles helps the framework scale to many workers and 1 helper.

As the framework can scale to 1 worker and many helpers, and to many workers and 1 helper, the framework can scale to many workers and many helpers. As the framework can scale to 0:0, 1:0, 0:1, 1:many, many:1, and many:many worker-helper relationships, the framework provides scalability.

In the same fashion as scalability, the other requirements benefit from the modular characteristics of each layer. By considering the most basic blocks of an interaction, you can address any concern more efficiently. For example, when we want to ensure trust, separating elements into only those who require the trust we are seeking, allows for the saving of resources and lowering the complexity. It is for this approach that we allow for more accessible requirements.

V. CASE STUDY

If someone is experiencing a cardiac arrest, a bystander would call emergency services. The answering dispatcher would ask for the bystanders location and for details about the emergency situation, and begins to help the bystander that starts performing cardiopulmonary resuscitation (CPR). Through our proposed framework we show how this CPR administration scenario of mobile health can be supported in a collaborative, M2M environment using a smartphone.

Applying our framework to this example, the Services layer contains the CPR task. The Roles layer includes the bystanders as workers, and the physicians, paramedics and dispatchers are the helpers. Some select Human/Computer Protocol elements we considered were a helper giving feedback to a worker and a hierarchy of many helpers. The Underlying Technologies include an Android application to collect and compute CPR statistics on the smartphone, a server implemented via Node.js, and web sockets secured by TLS. Figure 2 shows the relationships between elements of the CPR administration case study.

A. Implementation

Node.js standalone packages are available for Windows, Mac OS X, and Linux operating systems, effectively allowing our server software to run on most desktops, laptops, and servers. In addition, the Node.js asynchronous, event-driven programming framework makes it easy to scale to multiple clients and helpers. The operator portal uses standard HTML5 and JavaScript libraries, allowing dispatchers to use the portal on any computer with a fairly recent web browser. Our mHealth system can provide services such as CPR instruction and monitoring. In the context of the CPR monitoring portion

of our mHealth application, the mobile devices transmit the current time, date, and location to dispatchers and physicians, along with the depth and frequency of CPR compressions. The operator portal charts the depth and frequency information, and displays the clients location on a map so that dispatchers and physicians can advise clients effectively from remote locations. The mobile devices can detect connection losses and reconnect as needed until the mHealth application is stopped. It is important to note that can we administer CPR by anonymizing the personal details, and just by extracting pulse and CPR information from the data. Figure 3 contains a screenshot of the mobile CPR monitoring application. Figure 4 depicts the dispatcher protocol interface. Figure 5 is a sample report card generated from a CPR session. This report card gives the helpers feedback on how well the workers performed CPR. The report card assists the helpers in guiding workers on how to perform the High-Quality CPR [13].

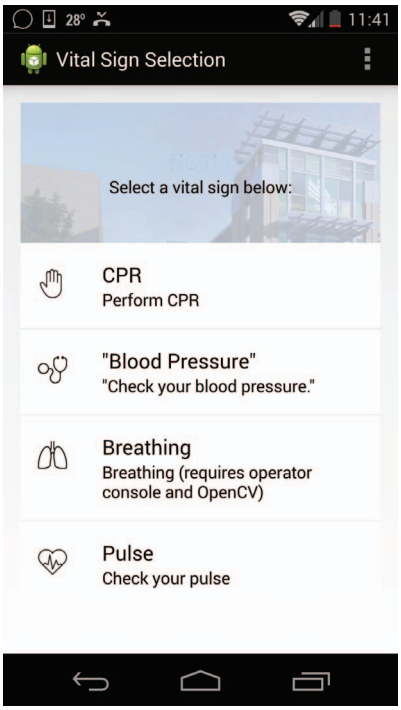

Figure 3. Screenshot of the mobile CPR monitoring application

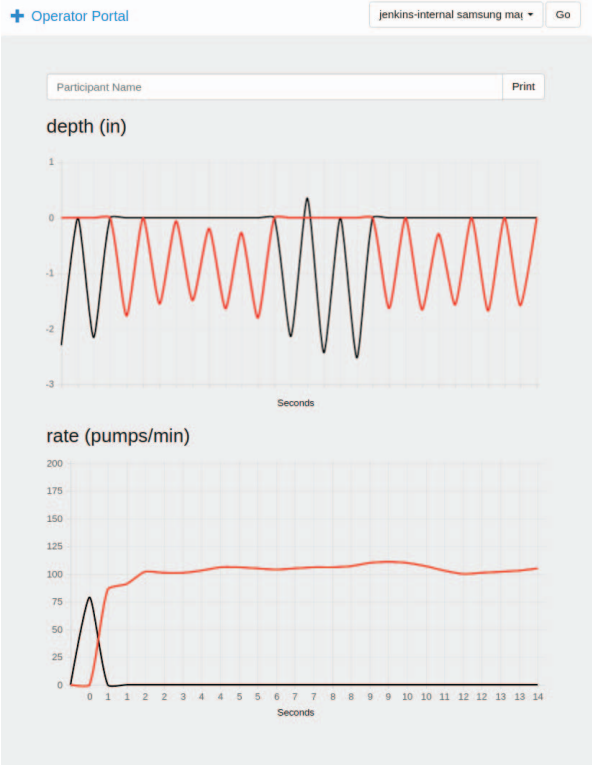

Figure 4. Example screenshot of a dispatcher protocol interface

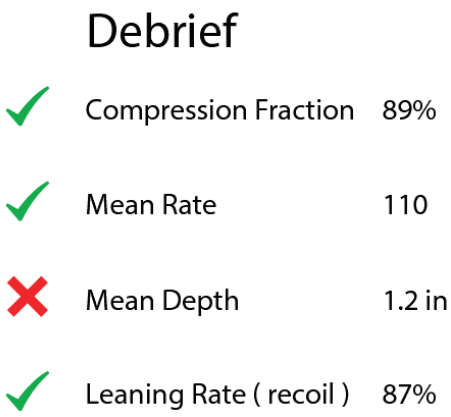

Figure 5. A sample report card from a CPR session to assess quality of CPR

To preserve anonymity, workers and helpers use the Underlying Technologies layer to establish a connection to an intermediary, a centralized server. When the connections to the server are established, workers use the Human/Computer Protocols layer to send device identifiers to the server. Then, the workers begin performing the tasks described in the Services layer. When desired, workers would use the Human/Computer Protocols layer to send service-specific information to the server. The server will forward any service-specific information to the helpers that are assisting the worker.

After the helpers connect to the server, helpers use the Human/Computer Protocols layer to retrieve lists of workers from the central server. When a helper decides which workers to assist, the helper then uses the Human/Computer Protocols layer to inform the central server of the decision. When a worker is done working, the worker performs any actions required by the Roles layer, and then uses the Human/Computer Protocols layer to send a termination message to the server. The server uses the Human/Computer Protocols layer to forward the termination message to the helpers that are assisting the worker.

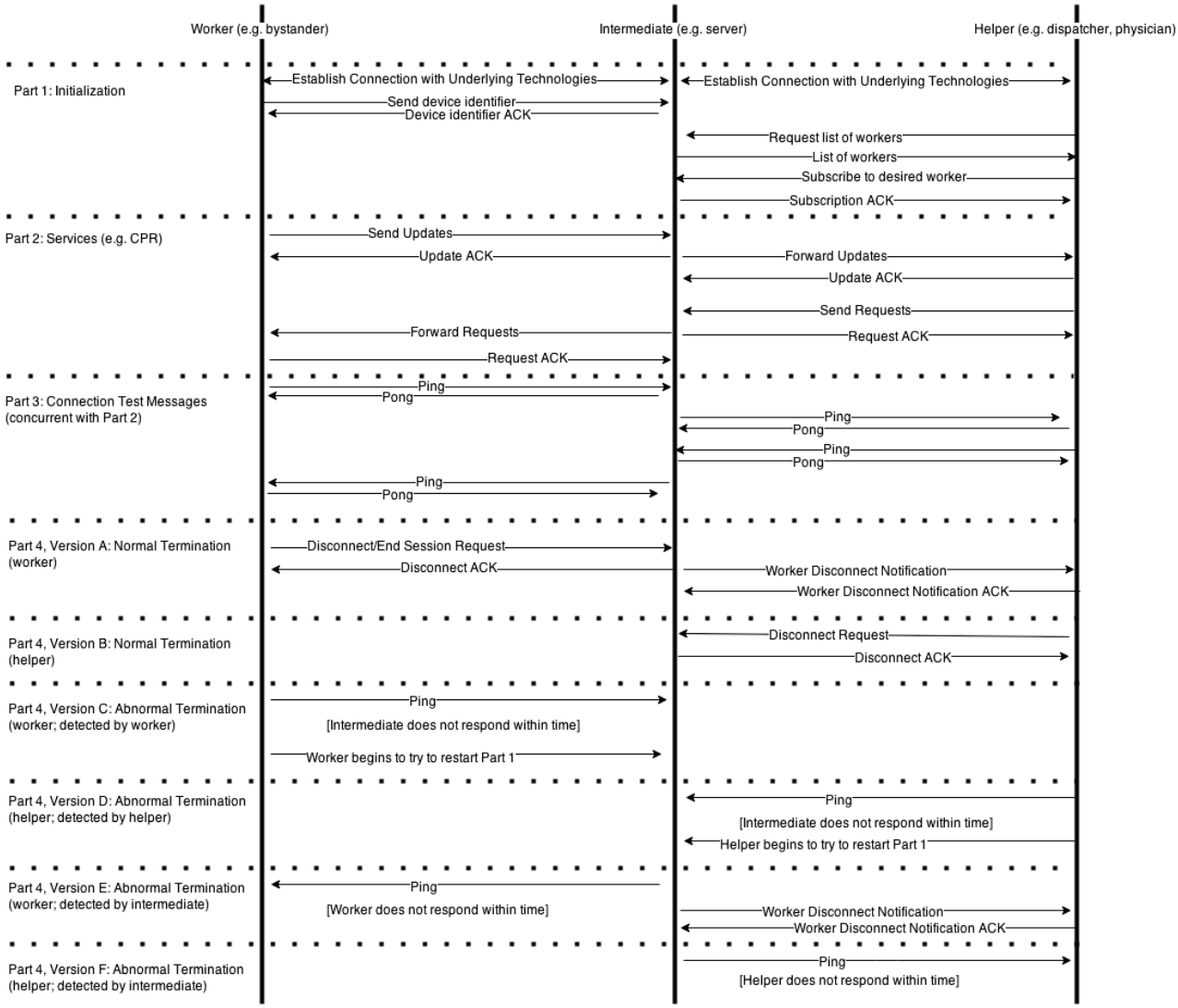

Figure 6. The event flow of the architecture for secure collaboration

If a helper decides to stop assisting a worker, the helper performs any actions required by the Roles layer, and then uses the Human/Computer Protocols layer to send a termination message to the server. This termination message is not forwarded to workers as the workers do not need to know what helpers are assisting them. If a worker detects a lost connection, the worker will continue performing the tasks described in the Services layer, while trying to reestablish a connection to the server through the Underlying Technologies layer. When the server detects the lost connection, the server will notify the registered helpers through the Human/Computer Protocols layer.

If a helper detects a lost connection, the helper will try to reestablish a connection to the server through the Underlying Technologies layer. As the workers do not need to know what helpers are assisting them, the server will not notify the workers that a helpers connection was lost. This preserves as much anonymity as possible, as the only information transmitted is the information required for the helpers to assist in providing services, and the credentials of the workers and helpers. Workers do not receive the helpers credentials. The credentials for both workers and helpers can be as minimal as device ID numbers and device names. Since the workers and helpers detect and try to reestablish lost connections, the connections are persistent. The server provides scalability and helps establish trust between the workers and helpers. Workers and helpers independently verify the trustworthiness of the server, and the server verifies the trustworthiness of the workers and helpers. The server also allows one helper to assist many workers, one helper to assist one worker, and many helpers to assist one worker. Workers continue performing their tasks if helpers are not available. Figure 6 shows the message flow between the worker, server and helper.

B. Performance

We installed our CPR application on two iPhones and two Samsung Galaxy Nexus phones, and used three laptops as dispatcher workstations. We confirmed that the server can support forwarding updates from multiple phones to a dispatcher (using tabs in the dispatchers browser), updates from a single phone to multiple dispatchers, and updates from multiple phones to multiple dispatchers.

VI. Limitations and Challenges

Trust: In an implementation of our framework, developers and system administrators may want to configure the intermediates such that only pre-authorized helpers are allowed to connect as helpers. For some applications, this form of access control may also be applicable for workers. However, for emergency applications such as our CPR administration case study, this form of access control would only be applicable to helpers.

Scalability: The scalability provided would vary based on specific implementation details, such as the processing power and bandwidth of the workers, helpers, and intermediates. System administrators may wish to cluster several intermediates to increase scalability.

Anonymity: The level of anonymity provided would vary based on the specific implementations of the layers of our framework.

VII. Conclusion

We have applied the framework described above to the field of mobile healthcare (mHealth). Specifically, we have used this framework to allow bystanders to perform cardiopulmonary resuscitation (CPR) using a smartphone application and additional assistance from physicians and emergency services dispatchers. Thus, we developed an adaptive framework and set of protocols that lends itself to secure collaboration. Achieving this was done by defining basic requirements such as trust, scalability, and anonymity as well as classifying an framework in which to support these requirements. Through the steps shown here, many collaborative M2M situations can organize themselves to maintain essential features while providing the security they require.

VIII. Future Work

The RMC protocols could be applied to the M2M framework presented in this paper using concepts from RFC 4353 [a]. In the M2M framework, the server would include conference policy, conference notication, focus, and mixer modules as described in the RFC. The worker would be the victim or bystander in the emergency situation that has a mobile device with a RMC-capable SIP client. In the application of the M2M framework, a helper would be an emergency services dispatcher or a physician. Applicable human protocols would include dispatcher-victim and physician-patient protocols. When a worker establishes a connection to the server, the server would create a new SIP conference for the worker. To assist a worker, a helper would join the workers SIP conference.

[a]https://tools.ietf.org/html/rfc4353

Acknowledgments

This work is partially supported by the National Science Foundation under grants CNS -0751205, CNS-0821736 and CNS-1229700. We acknowledge Jake Singer for his contributions to the CPR case study.

References

[1] L. M. Camarinha-Matos and H. Afsarmanesh, "Collaborative networks: a new scientific discipline," *Journal of intelligent manufacturing*, vol. 16, no. 4-5, pp. 439–452, 2005.

[2] T. Peng, C.-H. Chi, A. Chiasera, G. Armellin, M. Ronchetti, C. Matteotti, C. Parra, A. O. Kashytsa, and A. Varalta, "Business process assignment and execution in mobile environments," in *Collaboration Technologies and Systems (CTS), 2014 International Conference on*. IEEE, 2014, pp. 267–274.

[3] B. Solomon, D. Ionescu, C. Gadea, S. Veres, M. Litoiu, and J. Ng, "Distributed clouds for collaborative applications," in *Collaboration Technologies and Systems (CTS), 2012 International Conference on*. IEEE, 2012, pp. 218–225.

[4] B. Panja, D. Fattaleh, M. Mercado, A. Robinson, and P. Meharia, "Cybersecurity in banking and financial sector: Security analysis of a mobile banking application," in *Collaboration Technologies and Systems (CTS), 2013 International Conference on*. IEEE, 2013, pp. 397–403.

[5] A. L. Brooks and E. Brooks, "An internet of things resource for rehabilitation," in *Collaboration Technologies and Systems (CTS), 2014 International Conference on*. IEEE, 2014, pp. 461–467.

[6] S. B. Grimsmo, "Reliability issues when providing m2m services in the internet of things," 2009.

[7] J. Latvakoski, M. B. Alaya, H. Ganem, B. Jubeh, A. Iivari, J. Leguay, J. M. Bosch, and N. Granqvist, "Towards horizontal architecture for autonomic m2m service networks," *Future Internet*, vol. 6, no. 2, pp. 261–301, 2014.

[8] Z. Fan, R. Haines, and P. Kulkarni, "M2m communications for e-health and smart grid: an industry and standard perspective," *Wireless Communications, IEEE*, vol. 21, no. 1, pp. 62–69, 2014.

[9] S. Akter, P. Ray *et al.*, "mhealth-an ultimate platform to serve the unserved," *Yearb Med Inform*, vol. 2010, pp. 94–100, 2010.

[10] A. v. Heerden, M. Tomlinson, and L. Swartz, "Point of care in your pocket: a research agenda for the field of m-health," *Bulletin of the World Health Organization*, vol. 90, no. 5, pp. 393–394, 2012.

[11] M. Mintz-Habib, A. Rawat, H. Schulzrinne, and X. Wu, "A voip emergency services architecture and prototype," in *Computer Communications and Networks, 2005. ICCCN 2005. Proceedings. 14th International Conference on*. IEEE, 2005, pp. 523–528.

[12] V. Chandrasekaran, R. Dantu, and K. P. Subbu, "Socio-technical aspects of remote media control for a ng9-1-1 system," *Multimedia tools and applications*, vol. 62, no. 3, pp. 733–759, 2013.

[13] P. A. Meaney, B. J. Bobrow, M. E. Mancini, J. Christenson, A. R. de Caen, F. Bhanji, B. S. Abella, M. E. Kleinman, D. P. Edelson, R. A. Berg *et al.*, "Cardiopulmonary resuscitation quality: improving cardiac resuscitation outcomes both inside and outside the hospital a consensus statement from the american heart association," *Circulation*, vol. 128, no. 4, pp. 417–435, 2013.