Secure Information Flow Awareness for Smart Wireless eHealth Systems

Stefan Pfeiffer, Sebastian Unger, Dirk Timmermann
Institute of Applied Microelectronics and Computer Engineering
University of Rostock
{name.surname}@uni-rostock.de

Andreas Lehmann
Institute of Computer Science
University of Rostock
{name.surname}@uni-rostock.de

***Abstract*—Usually, safety and robustness are considered as the most important issues when designing and developing smart and wireless eHealth systems for ambient support of patients and physicians. However, information security is an indispensable design criterion due to the sensitive and critical nature of aggregated information from eHealth applications. Therefore, preventing undesired information leaks and illegal information flows is crucial for designing smart mHealth and mobile medical assistant systems. Due to the rapidly increasing complexity of these systems, finding information flow violations manually is nearly impossible. That is why there is a need for formally-based and tool supported verification methods of security properties.**

I. Introduction

During the last years, both industry and research communities are witnessing a growing interest in the technological evolution of electronic health systems (eHealth) [1]. The idea of smart mobile device enhanced eHealth systems (so called mHealth systems [2]) addresses the change of how we access healthcare information, bringing healthcare services directly into the personal space of the users [3]. This becomes even more important considering the weaker sections of the society including disabled, elderly, and chronically ill patients [4]. Smart mHealth systems can directly improve the quality of medical attendance by, e.g., enabling remote diagnosis and telemonitoring allowing for post-surgery rehabilitation and permanent vital sign supervision in familiar environments. However, sharing sensitive patient data in a large distributed and heterogeneous environment inherently raises a plethora of security and privacy risks [5]. Because of the rapidly growing complexity of these systems identifying non-functional security flaws in mHealth workflows manually is impossible and therefore, needs automation and tool support properly founded on formal methods. The sound verification of security requirements is an essential precondition for provable secure and therefore certifiable mHealth systems.

In this paper we present a model checking approach based on annotated Petri nets to verify the adherence of security properties and policies in mHealth based workflows. Our approach contributes to mHealth security by complementing the current mechanism-centric security research by providing a verifiable property-centric security perspective. Furthermore, we show how applying our approach can help analyzing existing and developing new *certifiable information flow secure smart mHealth systems*, guaranteeing a reasonable level of security in the large.

The remainder of this paper is structured as follows. Section II gives an overview of the current security research in mHealth environments and explains, why the mechanism-centric security perspective is insufficient for designing reliable secure mHealth systems. Section III introduces the notion of information flow awareness. To demonstrate the capabilities of our verification framework, we provide an mHealth scenario with a layered perspective in Section IV, leading to a classification of information flow violations typical for mHealth environments in Section V. Section VI introduces our verification framework based on annotated Petri nets and non-interference properties. Here, the power of formal verification is demonstrated using one of the provided workflows from Section IV. Finally, the paper concludes with a summary and short outlook in Section VII.

II. Related Work

Since mHealth systems are inherently dealing with sensitive patient data in large distributed environments, many researchers focus their work on security aspects in mHealth environments. To our best knowledge, current research is primarily based on developing new or applying existing security mechanisms to establish confidentiality and integrity of medical data. For instance, Deng et al. [1] develope a secure cross-context architecture architecture to manage distributed personal e-Health information, whereas MacDonald [6], Dmitrienko et al. [7] and Elmufti et al. [3] provide new methods and architectures for authentication. Furthermore, new secure access control architectures [7], [5], mHealth adapted secure communication protocols [8], and access control mechanisms, based on public key infrastructures for mHealth environments [9] have been developed.

However, primarily focusing on confidentiality aspects, a mechanism-centric view of security typically results in the translation of extensional, non system-specific confidentiality requirements into ordinary access control models. Thereby, this fact is neglecting that access control models are insufficient to regulate the propagation of information after it has been released for processing [10]. Consequently,

confidentiality can also be broken by the flow of information through a process. Neither access control mechanisms nor encryption provide complete solutions for protecting confidentiality [10].

III. Information Flow Awareness

The view of secure systems has traditionally been dominated by security-blackbox-thinking, considering the interior of a blackbox as trustworthy and the exterior as potentially hostile. Consequently, prior research focused on mechanisms securing this sharply defined borders by, e.g., restricting access to the interior and encrypting communication channels between the interior and the exterior (see related work). However, this sharp definition is blurred by aspects like networking, mobility, and dynamic extensibility. From a mechanisms-centric security perspective, applying security mechanisms does not prevent sensitive information from being unintentionally propagated. So, information flow control is one key aspect to design inherently secure systems, thus providing a powerful abstraction to certify confidentiality and integrity properties [11].

Dealing with sensitive and private data in the medical sector, upcoming mHealth infrastructures have to take information flow control into account. The awareness of where critical information is propagated to and the preventing of undesired information leaks is crucial for designing smart mHealth and mobile device supported medical assistant systems guaranteeing a certifiable level of confidentiality.

IV. Mobile device supported eHealth Scenario

The verification framework we present is an approach for verifying the absence of information flow violations and information leaks in medical workflow models. A medical workflow hereby is a sequence of concatenated medical tasks leading to the desired result, e.g., defining the emergency doctors behavior in case of emergency. In most cases, workflows abstract from technological details, focusing on the relevant task sequence. That's why we separate an mHealth environment into two different layers, an infrastructure layer, and a process definition layer. The technological environment consisting of smart mobile devices, clinical IT-infrastructure, and the communication network is encapsulated within the infrastructure layer. The process definition layer defines abstract medical workflows based on the underlying infrastructure layer. For specifying such medical workflow models, process specification languages, e.g., the Business Process Model and Notation language (BPMN) [12], can be used. The process designer has to decide on the appropriate level of model abstraction, raising it as high as possible hiding unnecessary, but still containing all relevant details for verifying the absence of information flow violations. Consequently, information flow policies, defining what kind of information flow is permitted, have to be defined in the process definition layer.

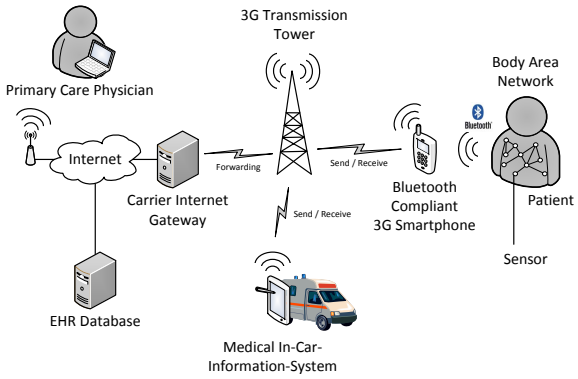


Fig. 1. Scenario: Mobile eHealth Monitoring Infrastructure

For presenting our verification approach, we will refer to the following mobile device supported eHealth scenario.

A. Infrastructure Layer

The infrastructure layer in our scenario is a technology enhanced version of the agent-based mobile health monitoring system by Chan et al. [13] (see Fig. 1). In addition to the original version, the patient is equipped with several bluetooth compliant monitoring devices and sensor nodes, spanning a sensing body area network (BAN). The sensing BAN has to fullfill the following tasks: monitoring of blood pressure and heartbeat, indoor localization and fall detection. The BAN is connected via Bluetooth to the patients personal Bluetooth compliant smartphone, which itself transmits the data over the GRPS/3G network to the Carrier Internet Gateway, providing the highest level of mobility. An electronic health record (EHR) database is connected as a web service to the Internet, providing access to the private and sensitive medical history of patients. For simplification, an EHR record in this scenario consists of entries for drug intolerances and medicamentous specialities of patients only. Additionally, an emergency physician is equipped with a mobile medical in-car-information-system, providing additional information for the current case of emergency enabling patient optimized first aid (e.g., with respect to drug intolerance).

B. Process Definition Layer

Based on the infrastructure layer, two constructed workflow examples defined in the process defintion layer are provided. Both examples are leading, for a better understanding of information flow problems, to obvious and therefore easy to see information flow violations. For modeling mHealthcare workflows, we use BPMN.

1) Emergency EHR Database Access: The first workflow example considers an emergency situation and applies the following information flow policy defined in the process definition layer:

- *Access to the EHR database is restricted to the patient and the primary care physician only.*

Every other access results in a violation of this security policy. Therefore, applying the information flow policy, access to the EHR database is only permitted to the patient (record owner) and the primary physician in charge. Workflows that allow access to the EHR records not in charge of the patient or the primary physician (e.g., unconscious by design) represent a discrepancy between the information flow policy and the actual workflow model. Consequently, this results in an information flow violation.

The fall detection sensors in the sensing BAN of a patient participating in a programm "rehabilitation at home after surgery" detected that the patient fell on the ground from where he did not stand up again. Together with a detected drop in blood preasure, implying that this fall possibly leads to life threatening bleedings, an emergency call is sent autonomously, containining the current location and unique identification number of the patient. An emergency doctor receives that call and immediately starts driving to the transmitted emergency location. Meanwhile, his medical in-car-information-system accesses the central EHR database with an emergency permission and requests the drug intolerance and medicamentous speciality entry from the patient checking for potential first aid drug intolerance. With this information, the emergency doctor provides a personalized first aid taking special characteristics and drug intolerances of the casualty into account. The relevant parts of the workflow for accessing the EHR database in case of emergency modeled in BPMN are shown in Fig. 2.

The information flow violation here can obviously be recognized by simply looking at the snippet. Even with a technological permission (access control) for accessing the EHR database in cases of emergency, the information flow policy defined for that specific example has been violated by design. We will use this example to demonstrate our developed methodology to automatically verify the existence or absence of information flow violations. Real world examples in fact are far more complex and opaque, not obviously revealing information flow violations. Thus, this example is for demonstrating purpose only.

2) Untrusted System Component: A second simplified workflow example avails a non-trustworthy smartphone from the infrastructure layer by means of a compromised and therefore hostile operating system. The workflow is as follows: Vital signs are permanently measured by the sensing BAN and transmitted via a secured Bluetooth connection to the patients smartphone. For transmitting the data over the 3G network, a system call to the untrustworthy operating system has to be done. Instead of directly sending the data, a compromised sending routine in the operating system copies all data and sends it to a hostile attacker.

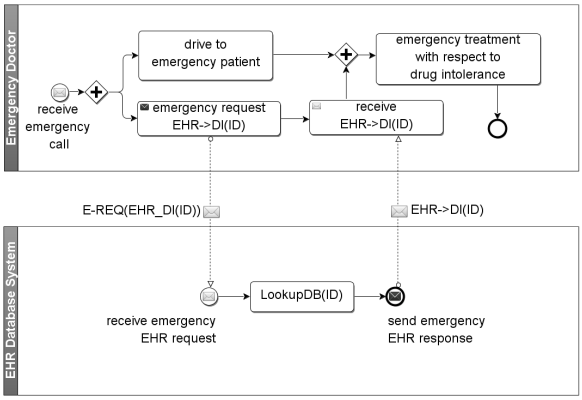


Fig. 2. BPMN Workflow Model: Emergency EHR Database Access

This workflow model seems to be constructed away from reality due to the very simplification for demonstrating purpose. However, statistics show that considering trojan horses and security vulnerabilities in current smartphone operating systems turn these omnipresent mobile devices into sources of potential security leaks [14]. This fact raises questions about their suitability in mHealth environments dealing with sensitive and private data as proposed by, e.g., Nguyen et al. [15] and Chan et al. [13]. Especially, considering a future certification of reliable secure mHealth architectures and workflows, relying on untrustworthy mobile devices has to be avoided.

V. Classification of mHealth Workflow Information Flow Violations

As the provided mHealth workflow examples show, several principal classes of information flow violations exist in the sector of mobile device supported eHealth environments. We suggest as a first step a very coarse grained, but in the context of this publication sufficient classification, helping to decide wether dedected information flow violations are acceptable or not.

A. Emergency Information Flow Violation

Emergency information flow violations are violations that occur in cases of emergency, threatening the life of casualties. In such situations, we have a clear prioritization of the patients life over an information flow policy. If such a policy would lead to the degradation of the patients vital conditions or even to death, then information flow violations are acceptable.

B. Untrusted / Not Certified System Components

Due to the increasing technological progress of smartphones and tablet computers, they are part of a seamless smart device integration into mHealth architectures. While smartphones are very flexible and cost efficient computing devices, they generally do not not offer sufficient security mechanisms to protect the data they operate on [7] and thus, must be considered as not trustworthy. Integration of non-trustworthy devices into mHealth workflows introduces potential security

risks. Information flow violations introduced by untrusted and potentially hostile smart devices are not acceptable in the area of mHealth.

C. Information Flow Policy Violation by Design

A third flaw in the information flow policy is created by the missing information flow awareness and accidential design flaws in terms of information flow control by infrastructure and workflow designers. Designing workflows based on mHealth infrastructures inherently must take information flow control into account. Such information flow violations by design are not acceptable in the area of mHealth.

VI. Verification Framework

The precondition of certifiable mHealth workflows operating on a given infrastructure layer is a sound, precise, and tool supported verification to prove the absence of information flow violations. Founding such a verification framework on formal verification methods is essential to guarantee provable and therefore certifiable results. We developed a generic framework for the automatic verification of information flow violations based on Petri net models that can easily be applied to mHealth workflow models. Figure 3 gives an overview of the overall verification framework. Here, we use workflow specification languages (e.g, BPMN or WS-BPEL) to model mHealth based workflows. Based on clear transformation semantics (e.g., [16]), we can map these workflow models to an annotated Petri net model. The annotated Petri net is automatically extended with special patterns encoding the desired information flow property by our tool Anica. Together with an information flow policy, a model checker can be used, solving the corresponding verification problem (here: reachability). If information flow violations according to the specific policy are found, a witness path is generated. Assuming clear bijective transformation semantics, we can use this information to identify the violation withing the corresponding workflow model and identify undesired information leaks.

Considering a very restrictive information flow property, we are able to reveal all possible information flow violations in workflows. Based on the translation of a given BPMN mHealth workflow specification into a Petri net model and based on assignments of confidential tasks, we can automatically check the workflow model for information flow violations and thus verify the absence of them.

A. Petri net based Non-Interference Verification

Considering that the workflow model on investigation is split into two logical security domains (*high* for confidential, *low* for public), a flow happens whenever information meant to remain in the secret domain *leaks* to the public domain. Following Denning [11], a workflow model is assumed secure if it enforces *non-interference*. That is, the actions in the *high* domain do not produce an observable effect (*interference*) in the *low* domain. The assumption is that interferences open

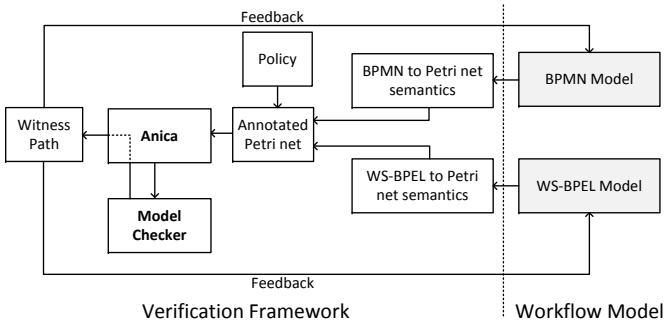


Fig. 3. Verification Framework

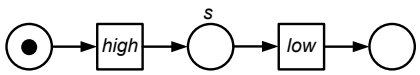
up the possibility to deduce information about confidential behavior. If exploited these so called *covert channels* violate the security requirements [17].

We consider the Petri net representation of workflow models as basis for our analysis. A Petri net is a mathematical modeling language for the description of distributed systems with a lightweight graphical notation, consisting of places, transitions, archs, and token. In contrast to other modeling languages such as BPMN, EPC and UML activity diagrams, Petri nets have an exact mathematical definition of their execution semantics, with a well-developed mathematical theory for process analysis.

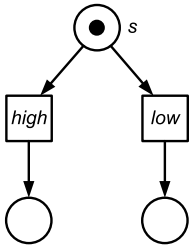
For the Petri net representation of workflow models, mappings from common modeling languages, such as WS-BPEL, BPMN, and EPC, exist [18]. Applying a BPMN to Petri net semantic allows us to describe ordinary mHealth workflow models using BPMN as high-level input language to our verification framework. To express the confidentiality requirements, we separate the BPMN tasks — modeled by Petri net transitions — into two logical security domains: *high* for confidential and *low* for public. An undesired leak and therefore an information flow violation happens whenever information meant to remain in the *high* domain *leaks* to the *low* domain.

The analysis of non-interference for such Petri net models is carried out with *positive place-based non-interference* (PBNI+) [19]. PBNI+ is an approach to encode and reason about *structural non-interference* (and hence information flow control) in Petri nets. The idea behind is that some specific places in the net encode different non-interference properties which represent leaks from the *high* to the *low* domain. By demonstrating the absence of such places in the net, one proves non-interference and therefore the absence of information flow violations.

Figure 4 depicts the two types of possible interference places, the causal case (a) and the conflict case (b). In the causal case, the *low* labeled transition t_2 can only fire after



(a) causal place



(b) conflict place

Fig. 4. Patterns for Potential Causal and Conflict Places s

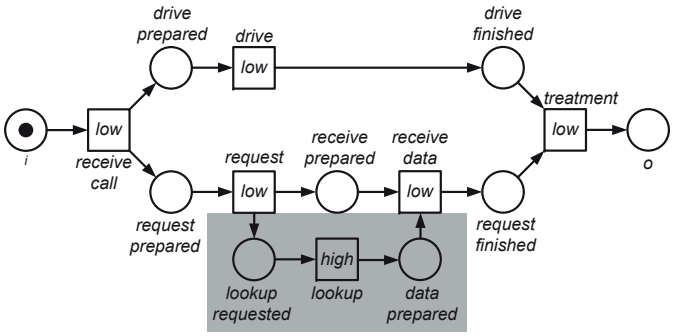


Fig. 5. Petri Net Model: Emergency Workflow (see Fig. 2)

the *high* labeled transition t_1 has fired, so the fact that t_2 (and its corresponding confidential task) has fired is leaked. In the conflict case the two transitions t_3 and t_4 are mutually exclusive, which means that from firing of the *low* labeled transition t_4 one may deduce that the *high* labeled transition t_3 has not fired. Both cases can be expressed as triple (s, h, l) of a place s, a *high* labeled transition h, and a *low* labeled transition l.

An annotated Petri net is secure in terms of PBNI+ if it contains no such places. Although it looks like a structural property, the behavior of the net needs to be considered to decide on PBNI+. Therefore, we first perform some structural checks to decide whether any of the two PBNI+ patterns exists in the Petri net (if not, the net is secure). Otherwise, we need to check all possible structural violations (also called *potential places*). As mentioned before, we have to examine the behavior of the net (i. e., its state space). These checks can be very expensive in terms of space and time (PSPACE complexity). However, based on our previous work [20], these checks can be expressed as independent reachability problems instead of an examination of the whole state space. Therefore, all checks can be done locally for each specific triple (s, h, l), where s is a potential violating place, h refers to the *high* labeled transition (cf. 4) and l identifies the *low* labeled transition.

To express PBNI+ as independent reachability problems, we have to create a new extended Petri net, based on the given Petri net (the initial model), for each triple (s, h, l). The creation of these extended net is done by a tool we developed called Anica, the automated non-interference check assistant [21]. Afterwards, the reachability problem can be decided by every suitable model checker for Petri nets capable of solving reachability problems. In our toolchain, we suggest to use our low-level Petri net analyzer LoLA [22]. For further details, the interested reader is referred to [20].

For each active place correlating to each identified information flow violation, the model checker can provide a witness path. The witness path contains the sequence of Petri net transitions, which is necessary to reach this violation. As we assume a bijective (based on clear semantics) relation between the transitions and the tasks in our initial process model, we obtain a replicable sequence of tasks in the mHealth BPMN model, leading to the information flow violation and therefore security vulnerability of the specific workflow model.

Regarding the workflow example (see Fig. 2) from our mHealth scenario, Fig. 5 shows the corresponding Petri net description. Due to simplification, we applied a bijective 1-to-1 semantic, where each task is translated into a Petri net transition taking the inherent concurrency of the model into account. Due to the information flow policy depicted from the example, tasks from the security domain "Emergency Doctor" are labeled as public (*low*) and tasks from the security domain "EHR Database System" are labeled as confidential (*high*). Following the definition, every information flow from *high* to *low* (by means of from the EHR database system to the emergency doctor) violates the given information flow policy.

The PBNI+ check on the Petri net shown in Fig. 5 reveals one potential causal place and thus, one potential information flow violation. This place is *data prepared* as the transition *lookup* in its preset is *high* annotated and the transiton *receive data* in its postset is *low* annotated, which yield to one triple to check: *lookup, data prepared, receive data*. This triple is an active one (see [23] for further details) and thus, the place *data prepared* is an active information flow violation. With our model checking approach, we gain the witness path: *receive call, request, lookup, receive data*. This means, whenever this path is executed in the given process model, the occurence of the task *lookup* is leaked

and confidential information flows from confidential to public domain.

Based on the results of the verification process, the mHealth workflow designer can use the witness path to identify the information flow violation and redesigning the workflow model to prevent this undesired information leak or be at least aware of it.

VII. Conclusion

In this paper, we introduced a property-centric view of security in mobile device supported eHealth systems. We state that the classic mechanism-centric security perspective is not sufficient to create reliable and therefore certifiable secure medical workflows upon an mHealth infrastructure and argue that it needs to be complemented by a property-centric view and information flow control. We presented our verification framework for static workflow verification based on Petri nets and the PBNI+ property to prove the absence of information flow violations. Together with clear bijective transformation semantics mHealth compliant workflows can be translated into Petri net models. The verification results can be fedback into the mHealth workflow model, identifying all information flow violations with respect to a given information flow policy.

The current state of our verification framework has one drawback: It requires a complete confidentiality assessment. That said, if an information leak was detected, the assessment has to be manually corrected and re-checked. In [23] we propose to provide a characterization of all valid confidential assessments given a partial (or even empty) confidentiality assessment.

Even if property-centric security verification on mHealth environments just started out, we believe, that the effort on the overall security and its impact on verifiable and therefore certifiable mobile device supported eHealth systems will be significant. It will possibly initiate a new era of security and information flow awareness, necessary to create reliably secure systems.

Acknowledgment

This work was partially funded by the DFG (German research foundation) in the project WS4Dsec in the priority programme Reliably Secure Software Systems (SPP1496).

References

[1] M. Deng, D. D. Cock, and B. Preneel, "An interoperable cross-context architecture to manage distributed personal e-health information," in *Handbook of Research on Developments in e-Health and Telemedicine: Technological and Social Perspectives*. IGI Global, Inc., 2009, pp. 576–602.

[2] R. Martí, J. Delgado, and X. Perramon, "Security specification and implementation for mobile e-health services," in *Proceedings of the 2004 IEEE International Conference on e-Technology, e-Commerce and e-Service (EEE'04)*. IEEE Computer Society, 2004, pp. 241–248.

[3] K. Elmufti, D. Weerasinghe, M. Rajarajan, V. Rakocevic, and S. Khan, "Privacy in mobile web services ehealth," *Pervasive Health Conference and Workshops*, pp. 1–6, 2006.

[4] Schwaibold, M. Gmelin, G. von Wagner, J. Schchlin, and A. Bolz, "Key factors for personal health monitoring and diagnosis devices," in *Mobile Computing in Medicine*, 2002, pp. 143–150.

[5] L. Martino, Q. Ni, D. Lin, and E. Bertino, "Multi-domain and privacy-aware role based access control in ehealth," in *Pervasive Computing Technologies for Healthcare, 2008. PervasiveHealth 2008. Second International Conference on*, February 2008, pp. 131 –134.

[6] J. MacDonald, "Authentication considerations for mobile e-health applications," in *Pervasive Computing Technologies for Healthcare, 2008. PervasiveHealth 2008. Second International Conference on*, 30 2008-feb. 1 2008, pp. 64 –67.

[7] A. Dmitrienko, Z. Hadzic, H. Lhr, A.-R. Sadeghi, and M. Winandy, "Securing the access to electronic health records on mobile phones," in *Biomedical Engineering Systems and Technologies*, Springer-Verlag, Ed., 2011.

[8] M.Aramudhan and K.Mohan, "New secure communication protocols for mobile e-health system," *International Journal of Computer Applications*, vol. 8, no. 4, pp. 10–15, October 2010, published By Foundation of Computer Science.

[9] G. Kambourakis, I. Maglogiannis, and A. Rouskas, "Pki-based secure mobile access to electronic health services and data," *Technol. Health Care*, vol. 13, pp. 511–526, December 2005.

[10] S. Zdancewic, "Challenges for information-flow security," in *In Proc. Programming Language Interference and Dependence (PLID)*, 2004.

[11] D. E. Denning and P. J. Denning, "Certification of programs for secure information flow," *Communications of the ACM*, vol. 20, no. 7, pp. 504–513, 1977.

[12] Object Management Group, "Business Process Model And Notation (BPMN), Version 2.0," January 2011. [Online]. Available: http://www.omg.org/spec/BPMN/2.0/

[13] V. Chan, P. Ray, and N. Parameswaran, "Mobile e-Health monitoring: an agent-based approach," *Communications, IET*, vol. 2, no. 2, pp. 223–230, Feb. 2008.

[14] McAfee, Inc., "Mcafee threats report: Third quarter 2011," 2011. [Online]. Available: http://www.mcafee.com/us/resources/reports/rp-quarterly-threat-q3-2011.pdf

[15] M. T. Nguyen, P. Fuhrer, and J. Pasquier-Rocha, "Enhancing e-health information systems with agent technology," *Int. J. Telemedicine Appl.*, vol. 2009, pp. 1:1–1:13, January 2009.

[16] R. M. Dijkman, M. Dumas, and C. Ouyang, "Semantics and analysis of business process models in BPMN," *Information & Software Technology*, vol. 50, no. 12, pp. 1281–1294, 2008.

[17] B. Lampson, "A note on the confinement problem," *Communications of the ACM*, vol. 16, no. 10, pp. 613–615, October 1973.

[18] N. Lohmann, H. Verbeek, and R. M. Dijkman, "Petri net transformations for business processes – a survey," *LNCS ToPNoC*, vol. II, no. 5460, pp. 46–63, 2009.

[19] N. Busi and R. Gorrieri, "Structural non-interference in elementary and trace nets," *Mathematical Structures in Computer Science*, vol. 19, no. 6, pp. 1065–1090, 2009.

[20] R. Accorsi and A. Lehmann, "Automated and fast information flow analysis for business process models," 2012, unpublished manuscript available at http://www.informatik.uni-rostock.de/ al357/reader.pdf.

[21] Service Technology, "ANICA - [A]utomated [N]on-[I]nterference [C]heck [A]ssistant," July 2011. [Online]. Available: http://service-technology.org/anica

[22] K. Wolf, "Generating Petri net state spaces," in *ICATPN 2007*, ser. LNCS 4546. Springer, 2007, pp. 29–42.

[23] A. Lehmann and N. Lohmann, "Model support for confidential service-oriented business processes," in *Proceedings of the 4th Central-European Workshop on Services and their Composition, ZEUS 2012, Bamberg, Germany, February 23–24, 2012*, 2012.