Using Combination of MAPE-K and DSPL to Secure Smart Camera Networks

Mohamed AMOUD, Ounsa ROUDIES
SIWeb Team - École Mohammadia d'Ingénieurs (EMI)
Mohammed V University in Rabat. Morocco
amoudmohamed@gmail.com, roudies@gmail.com

Abstract

Smart Camera Networks (SCN) are becoming a fundamental piece of our intelligent cities, buildings and homes, progressively inserting themselves in our lives. Achieving security in such a dynamic and heterogeneous environment is a challenging task. This means that applications for Smart Camera Networks need mechanisms for self-adaptation and for self-protection based on the dynamic reconfiguration and adaptation of the algorithms used to provide security. In this paper, we propose a self-adaptive security solution for SCN based on the combination of the MAPE-K reference model to dynamic negotiate and deploy of security policies, and DSPL approach to reconfigure the security level of the applications at runtime and monitor the changes in the context. The novelty of our approach comes from dynamic negotiation of security policies and automatic reconfiguration of security level to instantiate the new security policies at runtime in order to be capable of deploying adaptive security mechanisms to satisfy different security needs at different conditions.

Keywords

Smart Camera Networks, MAPE-K, DSPL, Security policies, Dynamic adaptation, Dynamic reconfiguration, Self-protection.

1. Introduction

In many domains, systems need to run continuously and cannot be shut down for reconfiguration or maintenance tasks.

Smart Camera Networks are composed of heterogeneous devices. SCN work in dynamic environments with changing information about the relative camera location. Such devices must adapt their position and their function in response to changes and fluctuations in tasks and in the environment [3]. Dynamically changing SCNs are a challenge, as static security mechanisms are not able to offer an optimal security level for the varying situations. Moreover, it is impossible at design-time to anticipate all situations in which a SCN application will be utilised. These challenges cause a need for self-adaptive security, which is able to select security mechanisms and tune their parameters autonomously at runtime. On the other hand, Software development processes should provide security solutions that take into account the high heterogeneity and variability of SCN and applications and also the high variability of both security requirements and solutions.

Hence, this paper presents a methodology detailing key steps and activities to create an adaptive security solution based on the use of the MAPE- K reference model to dynamic negotiate and deploy of security policies by modeling secure interoperability policies with different constraints, and DSPLs approach to reconfigure the security level of the applications at runtime and monitor the changes in the context in order to be able to react to those changes. Finally, we have validated our work using a variability modelling technique: the Common Variability Language (CVL).

After the introduction background information is given. Section 3 describes the main challenges that need to be solved and presents the details of our approach. The evaluation of our approach is discussed in Section 4. A related work is presented in Section 5. Finally, the Conclusion and future work Section close the paper.

2. Background

2.1 MAPE-K Reference Model

In the IBM's Autonomic Computing vision [7], the autonomic behaviour is achieved by means of the MAPE-K reference model (Monitor, Analyse, Plan, Execute, and Knowledge). The principle of the Monitor phase is to collect information from the managed element, that is, the adapted software, and execution environment. The monitoring utilises sensors to collect relevant data. Therefore, the Plan recognises what has to be changed in order to achieve requirements and how to perform this change. The output of this phase is called an adaptation plan, which utilises self-configuring and self-optimisation. The Analyse phase combines the collected data and possible history data to reveal if requirements are not fulfilled, which causes an adaptation need. Consequently, the Analyse phase calls the Plan phase, which creates the adaptation plan. The adaptation plan contains a decision on how the software will be adapted. In order to create the adaptation plan, different algorithms or rules are utilised. Moreover, the Plan phase takes possible contradicting requirements into account as trade-offs. Finally, the Execute phase enforces the adaptation plan by means of effectors, which affect the managed element.

2.2 Dynamic Software Product Lines

Software Product Lines (SPL) have been used successfully in industry for building families of systems of related products, maximizing reuse, and exploiting their variable and configurable options.

New SPLs extend the concept of conventional SPLs by enabling software-variant generation at runtime and produce software capable of adapting to such fluctuations. In contrast with traditional SPLs, dynamic SPLs (DSPL) bind variation points at runtime, when software is launched to adapt to the current environment, as well as during operation to adapt to changes in the environment. Building a product line that dynamically adapts itself to changing requirements implies a deployment of the product configuration at runtime [16]. It also means that the system requires monitoring capabilities for detecting changes in the environment. As a response to these changes, the system adapts by triggering a change in its configuration, providing context-relevant services or meeting quality requirements. Dynamic software reconfiguration is concerned with changing the application configuration at runtime after it has been deployed.

3. Challenges and our approach

In this section we describe, in more detail, the main challenges that need to be taken into account in order to dynamically adapt and reconfigure the security level at runtime. We also indicate how our approach copes with these challenges [6] - [7] - [9] - [17].

3.1 Challenges

Challenge1: Develop secure applications that run on heterogeneous systems. The SCN are heterogeneous, and thus, the security solution deployed at each component needs to be adapted to the characteristics of every component.

Challenge2: Dynamic negotiation of security policies for SCN. The conditions of the running environment of the systems could change. Then, the security policies may need to be adapted to the new conditions. This adaptation requires a previous negotiation between the different parts of the application.

Challenge3: Monitor the changes in the context. These changes can be general or can be specific to security. The environment needs to be monitored to react to these changes.

Challenge4: Define a dynamic reconfiguration service to endow applications with self-protection. When the environment or the security requirements change at runtime, the system must be self-protected to maintain a proper and secure functioning.

Challenge5: Provide an efficient solution able to run on SCN. As SCN are composed of a large number of devices and applications, the self-protection mechanism should be efficacy and efficient as possible.

3.2 Dynamic Adaptation of Security

Based on the challenges described before, in this section we describe a methodology to realize an adaptive security solution at runtime that conforms to IBM's Autonomic Computing vision [7] and which is able to select security mechanisms and tune their parameters autonomously at runtime.

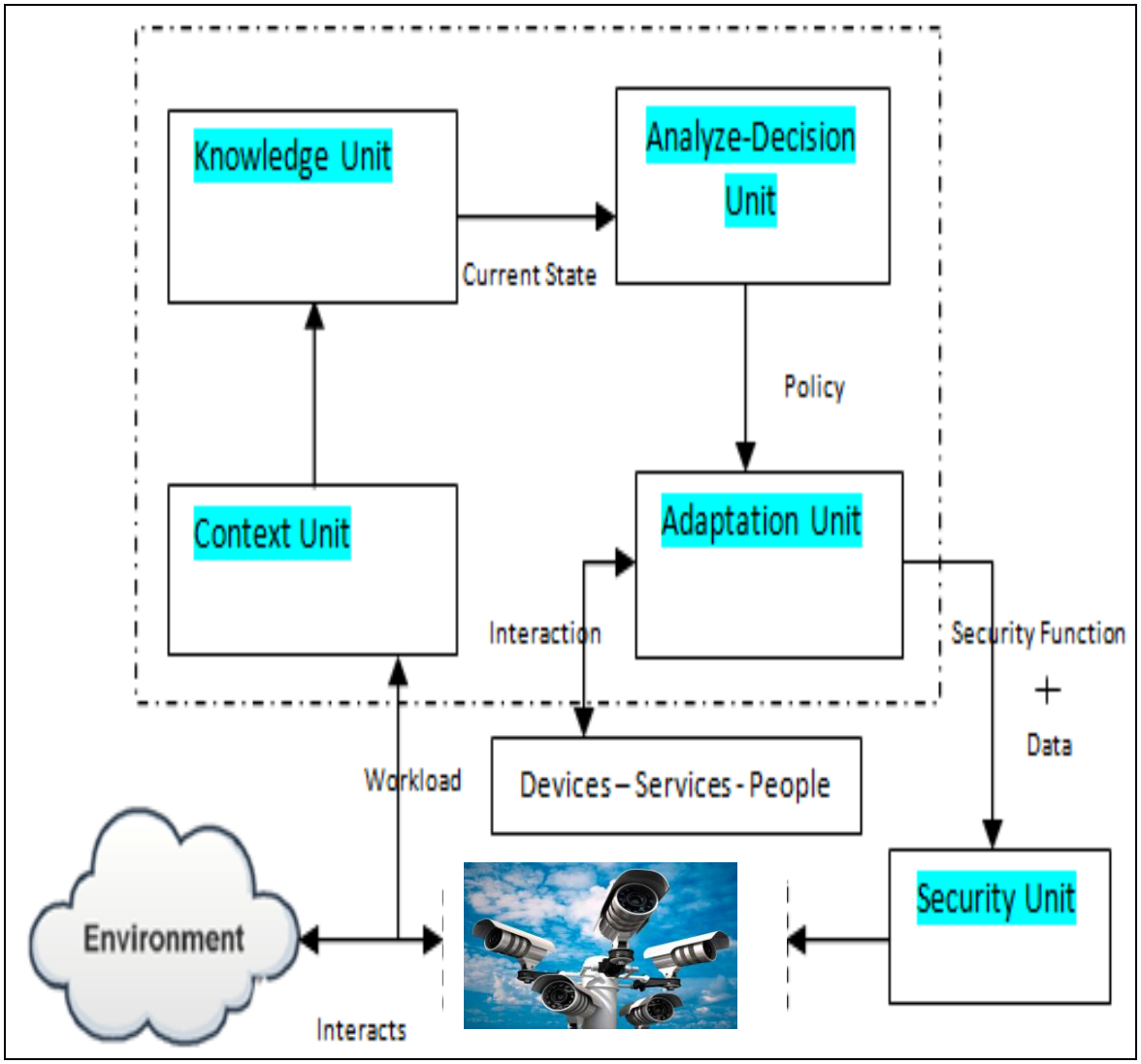

Figure 1.. Adaptive Security for Smart Cameras

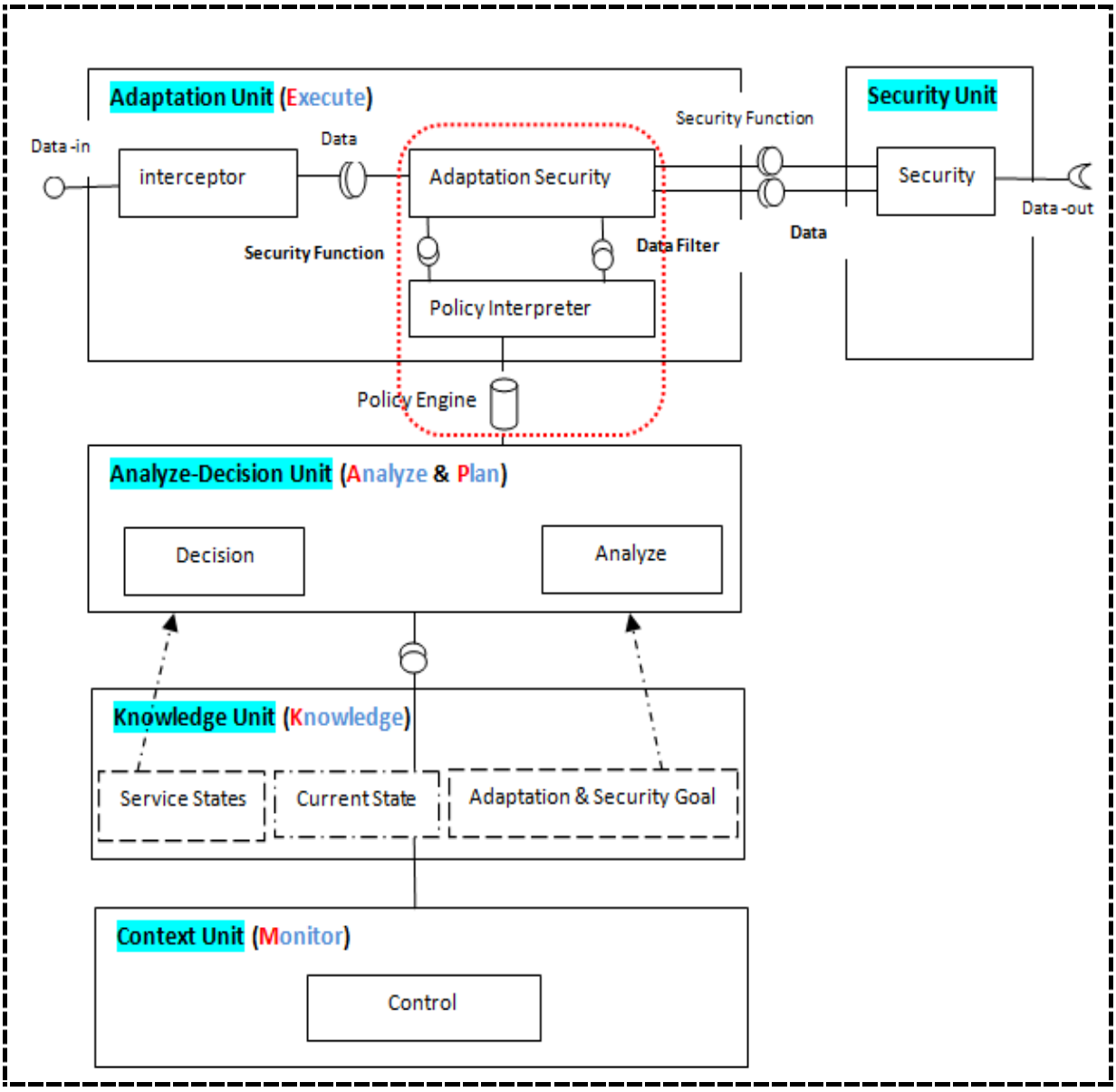

Figure 2. Adaptive Security Architecture

Figures 1 and 2 depict the components of an adaptive security service in dynamic systems which augment a standard security service with adaptive features and new activities supporting the runtime nature of systems assets and products [1]. Furthermore, the design is a realisation of the adaptive feedback loop of MAPE-K model.

- **Context Unit:** It is similar to the Monitor component in the MAPE-K Model. It enables the Adaptive Security Service to observe the environment context. Through runtime monitoring of the system context it enables the *Analyze-Decision Unit* to make informed decisions based on the *Current State* of the system.

- **Knowledge Unit**: The shared knowledge includes data such as: -The *Service States* file contains pre-computed data or a system model which represents the interrelationship between security and the environmental concern. - The *Current State* which allows the *Analyze-Decision Unit* to determine the future system state: the change in the relationship, if security is adapted. With reference to the *Service States* the *Analyze-Decision Unit* generates a *Policy* which satisfies the adaptation *Goal*.

- **Analyze-Decision Unit:** Taking the *Current State* into account the *Analyze-Decision Unit* is thus able to determine the future system state, i.e. the change in the relationship, if security is adapted. The *Analyze* component structures the actions needed to achieve goals and objectives. The *Decision* component determines if and how security should be adapted based on the adaptation *Goal*.

With reference to the *Service States* the *Analyze-Decision Unit* generates a *Policy* which satisfies the adaptation *Goal*.

- A**daptation Unit**: in some sort, it is the architecture control center. It is a unit whose role is paramount and central. It corresponds to the Executor service in the MAPE-K Model and is responsible for implementing and enforcing the security adaptation at runtime. It is involved in the adaptation process in two key areas : 1) intercepts the source of data in transit to the security unit by forming a checkpoint where security can be adapted; 2) it maps the given Policy rules to the Source Data indicating to the *Security Unit* which Security function to apply to which sets of *Data*.

The block consisting of the *Policy Interpreter* and the P*olicy Engine* modules decide whether or not a new security policy must be negotiated and the changes to be performed in the system security after the negotiation.

The *Policy Interpreter* parses the incoming policy updates, generating appropriately formatted *Security Function* and *Data Filter* pairs as input to the *Adaptation Engine*. Such parsing may include authenticating the sender; policy syntax checking and mapping policy rules to internal security algorithm and *Data filter* representations.

Then the *Adaptation Engine* applies the filter conditions specified in the *Data Filter* to the incoming data from the *Interceptor*. Once filtered, the Data is sent to the *Security Unit* specifying the appropriate *Security Function* to apply to the particular subset of data.

- **Security Unit**: takes source data as input, applies the relevant *security function* to the data and returns the transformed data.

Our methodology applies the whole MAPE adaptation loop for a Dynamic Adaptation of Security in SCN and defines each phase separately to successfully create an Adaptive Security Service.

3.3 Dynamic Reconfiguration of Security

DSPL offers a suitable development model for developing configurations of many independently developed systems which use services from and provide services to each other. In this way, dynamic reconfiguration is defined in terms of replacing the current variability model configuration with a new configuration in which the variation points have been re-bound to adapt them to a context change. By applying this approach, SCN are able to configure and reconfigure instances of the middleware, in order to solve the problems of modeling related to its heterogeneity and variability [18].

In addition to this, we use the Common Variability Language (CVL) to specify and resolve variability. The CVL variability models allow modeling the variability separately from the base model, but both the variability and the base models are connected and can be managed using the same tool (fig. 3). This CVL approach can automatically generate both the initial configuration and also the successive adapted configurations.

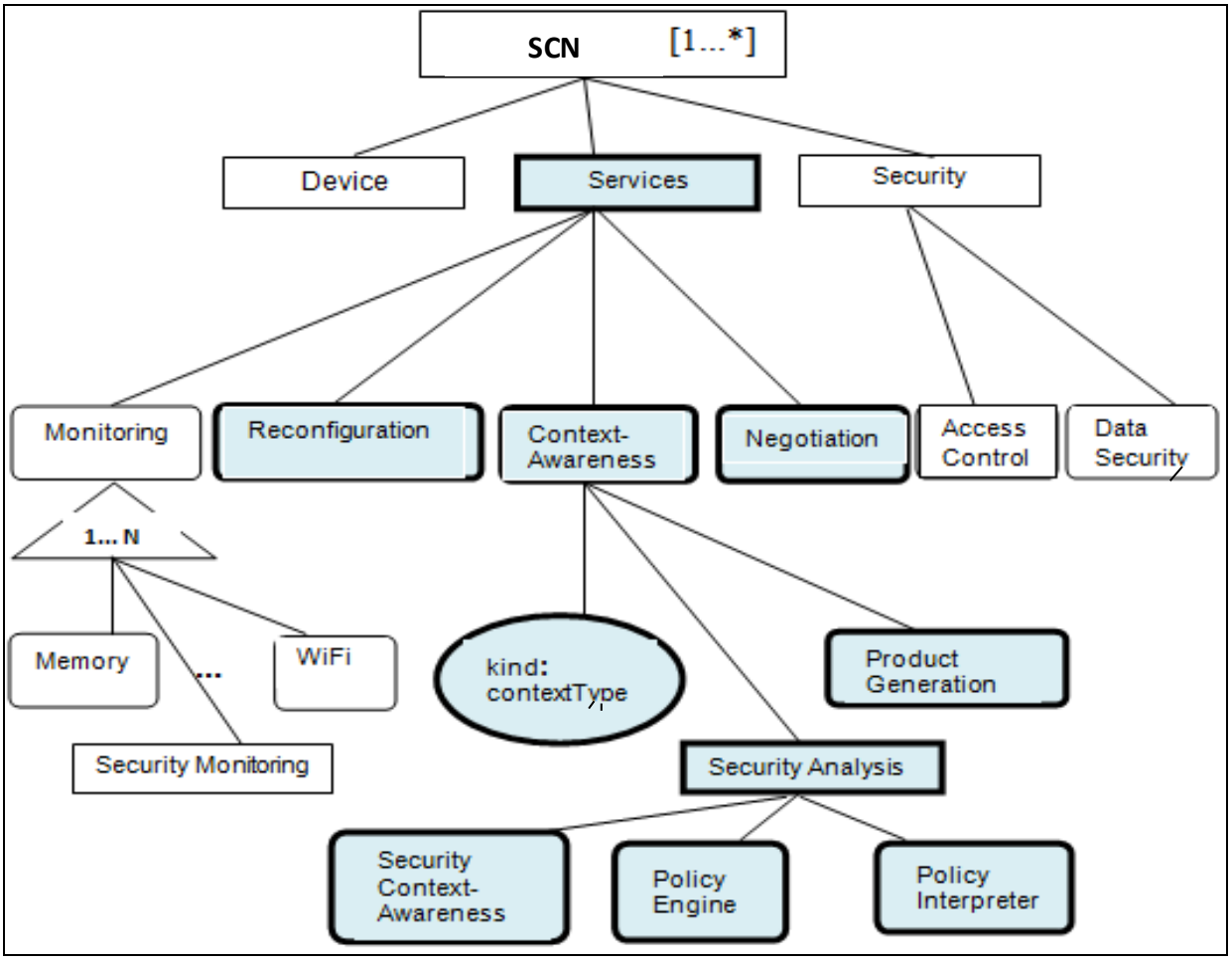

Figure 3. Dynamic Reconfiguration of Security Variability Model

Using Variability Models at run-time guides the choice of system variants in response to context changes. These operations indicate how system components should be reorganized for the reconfiguration in order to move from one configuration of the systemto another configuration.

Context awareness: The context awareness feature is represented by two elements. One is the kind of context of which a certain application is required to take care. The other is the list of changes to be executed to generate a new product when a context modification implies the system's reconfiguration. This is carried out by the security context awareness together with the policy interpreter and the policy engine modules. These modules decide whether or not a new security policy must be negotiated and the changes to be performed in the system security after the negotiation. Thus, a new feature called security analysis has been added as a child of the context awareness feature that includes the security context awareness, policy engine and policy interpreter children. Moreover, a new feature has been added as a child of the **product generation** feature to indicate that, in the case of security, the new product configuration needs to be generated, taking into account the information dynamically generated by the policy interpreter module. Thus, now, the reconfiguration plans could be generated taking as input the information generated by the policy interpreter module [20].

Reconfiguration: This module is in charge of reconfiguring the security of the system by instantiating the new negotiated security policy in all of the necessary SCN of the network. Since the context awareness module generates a reconfiguration plan with the same format for both generic and security-specific reconfigurations.

Negotiation: included at the same level as context awareness and reconfiguration services for the special requirements of the dynamic negotiation of security policies.

4. Evaluation and discussion

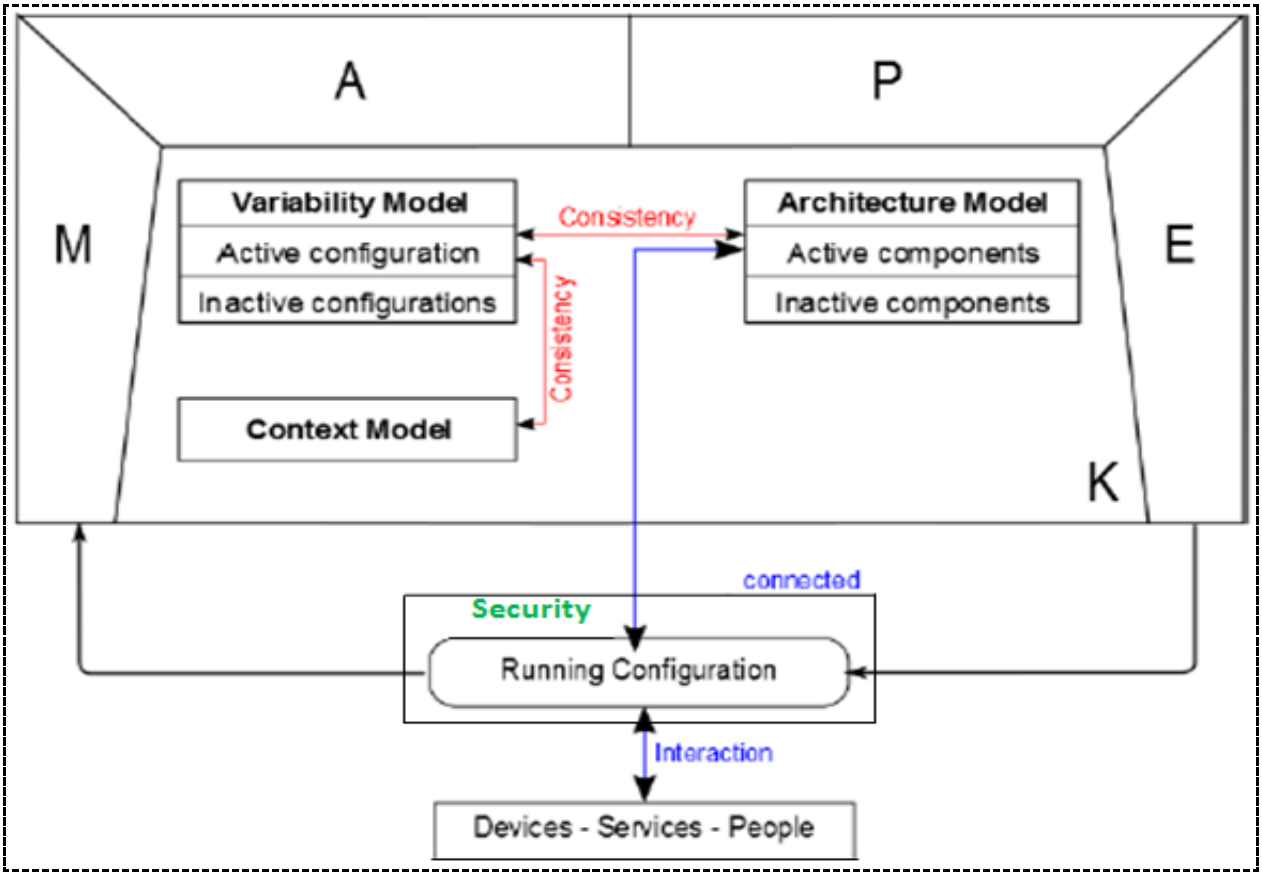

Figure 4. Security solution based on the combination of MAPE-K and DSPL

We have been combining both MAPE-K and DSPL to achieve adaptive security in SCN (fig. 4). Our research shows that variability models at run-time can assist a system to determine the steps that are necessary to reconfigure itself. DSPL development mainly intends to produce configurable products whose autonomy allow to reconfigure themselves and benefit from a constant updating at runtime in accordance with the changes in the environment. These updates enable systems to deal with context changes. We argue that a system can activate/deactivate its own features dynamically at run-time according to the fulfilment of context conditions.
When the security requirements change at runtime, the security policies are (re)negotiated. The negotiated security policy is then analyzed and interpreted by the policy engine and the policy interpreter modules. These modules are responsible for identifying changes in the security policy that require adapting the security concerns deployed inside the application dynamically, at runtime.

5. Related work

Few surveys from the adaptive security field already exist [11]-[12]-[21]. Hallsteinsen et al. [10] present the MADAM approach to building adaptive systems. They extend SPLs by adding the ability to automatically derive

changed configurations by monitoring the context, and to automatically reconfigure the security application while it is running. Nevertheless, they do not provide an implementation of a negotiation process as we do.

Hallsteinsen and al. [13], in MUSIC approach, propose to combine (1) DSPL Architectures and (2) Service Level Agreement (SLAs). SLA negotiation coordinates the configuration of a set of interacting systems. DSPL offers a suitable development model for developing configurations of many independently developed systems. Assuming a system of systems where all systems are built as DSPLs with utility and property predictor based decision models. They propose that each system is configured separately but the configuration is coordinated through SLA negotiation, but they do not propose a generic framework to provide context adaptation as our methodology does.

Parra and al. [14] argue that using an SPL paradigm to build context-aware systems based on SOA services, they propose a homogeneous Context- Aware Dynamic Service-Oriented Product Line (DSOPL) named CAPucine. Their goal is to define at the same time a service-oriented and context-aware product derivation that monitors the context evolution in order to dynamically integrate the appropriate assets in a running system, but no model-driven process has been completed.

The MAPE-K is commonly utilised as a reference model in different adaptation approaches [4], and thus, elements from the MAPE-K model are emphasised in this section. Elkhodary et al. surveyed three security adaptation approaches in [5] – namely Extensible Security Infrastructure [8], The Willow Architecture [2] and the Adaptive Trust Negotiation Framework. As the final conclusion, authors notice that any of these approaches support all security objectives but concentrate on specific and pre-selected objectives.

Elrakaiby and al. [15] present a policy-based approach for automating the integration of security mechanisms into Java-based business applications. They use security@runtime, a Domain-Specific Language (DSL), for the specification of security configurations based on authorization, obligation and reaction policies. Our approach is suitable for using security policies specified in any model since the mapping between the policies and the security functionalities is made at an abstract level of the variability model. We separate the monitoring of changes in the application and the integration of the security functionality following the MAPE-K loop, while they integrate the security functionalities in the same monitoring events.

M. Loughlin and al. [19] present current methodologies with privacy and trust perspectives and proposes a multi-layered security approach highlighting the need for a public key infrastructure layer associated with a reputation-based cooperation mechanism.

Many systems, architectures and frameworks currently exist to support such runtime decisions but the challenge we address in this paper lies in moving such a decision from a static design/deployment time decision to a dynamic runtime decision. This concept is at the heart of our methodology which takes legacy security systems and provides steps to create an adaptive security solution based on the use of the MAPE- K reference model to dynamic negotiate and deploy of security policies, and DSPLs approach to reconfigure the security level of the applications at runtime and monitor the changes in the context in order to be able to react to those changes, what constitutes the novelty of our contributions.

6. Conclusion and future work

This paper concentrated on self-protection solution for SCN based on the MAPE-K Model and DSPL paradigm. Hence, This approach allows us to: - build secure applications that run in heterogeneous conditions; - dynamically satisfy different security needs at different situations; - monitor the changes in the context in order to be able to react to those changes; - reconfigure the security level of the applications at runtime, and - perform the previous tasks and challenges using an efficient solution that is able to work at runtime.

As an emerging topic, we expect that promising new research will bring better and integrated solutions for adaptive security in Mobile Devices and Mobile Cloud Computing.

References

1. Capilla, R., et al., An overview of Dynamic Software Product Line architectures and techniques: Observations from research and industry. J. Syst. Software, 2014,
2. Knight, J.; Strunk, E. Achieving Critical System Survivability Through Software Architectures. In Architecting Dependable Systems II, Springer Berlin / Heidelberg, pp. 69–91, 2004.
3. J. C. SanMiguel, C. Micheloni, K. Shoop, ‘’Self-Reconfigurable Smart Camera Networks’’. Journal Computer, Volume 47, Issue 5, Pages 67-73, May 2014.

4. Yuan, E.; Malek, S. A taxonomy and survey of self-protecting software systems. In Proceedings of the IEEE Software Engineering for Adaptive and Self-Managing Systems, Zürich, Switzerland, pp. 109–118 4-5, 2012.
5. Elkhodary, A.; Whittle, J. A Survey of Approaches to Adaptive Application Security. In Proceedings of the Software Engineering for Adaptive and 65 Self-Managing Systems Workshop, Minneapolis, 20–26 May, IEEE, 2007, pp. 16–23.
6. Mónica Pinto and al.,'' Dynamic Reconfiguration of Security Policies in Wireless Sensor Networks'', Sensors 2015, 15, 5251-5280; doi:10.3390/s150305251, 2015
7. "An Architectural Blueprint for Autonomic Computing," June 2006. Available: http://www-01.ibm.com/software/tivoli/autonomic/pdfs/AC Blueprint White Paper 4th.pdf
8. R. Adler, D. Schneider, and M. Trapp. ''Development of safe and reliable embedded systems using dynamic adaptation''. Workshop on Model-Driven Software Adaptation, M-ADAPT 2007 at ECOOP, pp. 9- 14, 2007.
9. C. J. Lamprecht: '' Adaptive Security ''. PhD's thesis, Newcastle University, April 2012.
10. S. Hallsteinsen, E. Stav, A. Solberg, and J. Floch. Using product line techniques to build adaptive systems. Software Product Line Conference, 2006 10th International, pages 21-24, Aug. 2006.
11. A. Evesti and E. Ovaska :`` Comparison of Adaptive Information Security Approaches'', Hindawi Publishing Corporation, ISRN Artificial Intelligence, Volume 2013, Article ID 482949, 18 pages, 24 August 2013
12. S. Cheng, J. Zhang: ``Adaptive-ID Secure Revocable Identity-Based Encryption from Lattices via Subset Difference Method'', In Proceedings of the 11th International Conference on Information Security Practice and Experience (ISPEC 2015); 10 Feb 2015.
13. S. Hallsteinsen, S. Jiang. Dynamic software product lines in service oriented computing. In 3rd International Workshop on Dynamic Software Product, 2009.
14. Carlos Parra, Xavier Blanc, and Laurence Duchien. Context Awareness for Dynamic Service-Oriented Product Lines. Software Product Line Conference, 2009. SPLC 2009. 13th International, 24-28 Agust. 2009.
15. Elrakaiby, Y.; Amrani, M.; Le Traon, Y. Security@Runtime: A Flexible MDE Approach to Enforce Fine-grained Security Policies. Eng. Secur. Softw. Syst., 8364, 19–34, 2014.
16. D. Hausknecht. '' Variability-aware Data-flow analysis for Smartphone Applications''. Master's thesis, Faculty of Computer Science and Mathematics, University of Passau, 28 september 2013
17. Jesus Abelarde, '' Security Systems Engineering Approach in Evaluating Commercial and Open Source Software Products'', SANS Institute InfoSec Reading Room, January 25, 2016
18. B. Rinner, L. Esterle, J, Simonjan ''Self-Aware and Self-Expressive Camera Networks''. Journal Computer, Volume 48, Issue 7, Pages 21-28, July 2015.
19. M. Loughlin, A. Adanane, ``Privacy and Trust in Smart Camera Sensor Networks``, 10th/IEEE International Conference on Availability, Reliability and Security. Toulouse, France, 24-27 August 2015.
20. U. Ramachandran, L. Iftode, K. Hong,`` Large-Scale Situation Awareness With Camera Networks and Multimodal Sensing ``, Proceedings of the IEEE Volume: 100, Issue: 4, pp: 878 - 892, April 2012**.**
21. P. Li, L. Guo, M. Liu, ``Camera security network design and realization based on PCA facial recognition algorithms``IEEE/ICCICCT International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT) , India, 2016.

Biography

Mohamed AMOUD received the Master of Science degree in computer science from the University of Montreal, Canada in 2008, and the MBA degree in science and engineering from the School of Management (ESG UQAM), Montreal, Canada in 2010. He is currently working towards the Ph.D. degree in Ecole Mohammadia d'Ingénieurs (EMI), Mohammed V University in Rabat-Morocco. His research interest includes Computer Security, DSPL and Mobile Computing

Ounsa Roudies received the Ph.D. degree in computer science from the University of Paris VI, France, in 1989. She is a Professor of Computer Science in Ecole Mohammadia d'Ingénieurs (EMI), Mohammed V University in Rabat-Morocco; Co-Editor of the eti Journal. She leads the Siweb research team, conducting research in the areas of IS, Reuse, DSPL, Web services, CBR, Computer Security and Reliability, Composition, patterns, Quality.