IBIMA Publishing
Journal of Internet and e-Business Studies
https://ibimapublishing.com/articles/JIEBS/2022/277290/
Vol. 2022 (2022), Article ID 277290, 15 pages, ISSN: 2169-0391
DOI: 10.5171/2022.277290

Research Article

Architectural and Implementation Aspects of Security Mechanisms for Digital Electronic Integration Platforms

Jaroslaw Wilk[1] and Boleslaw Szafranski[2]

[1]SoftwareONE / Military University of Technology, Warsaw, Poland
[2]Military University of Technology, Warsaw, Poland

Correspondence should be addressed to: Jaroslaw Wilk; jaroslaw.wilk@softwareone.com

Received date:21 September 2021; Accepted date:16 April 2022; Published date: 18 July 2022

Academic Editor: Marcin Lawnik

Copyright © 2022. Jaroslaw Wilk and Boleslaw Szafranski. Distributed under Creative Commons Attribution 4.0 International CCBY 4.0

Abstract

Electronic integration platform environments are increasingly used as a mechanism for sharing and integrating e-services delivered to a recipient. The authors in their previous publications prepared a mathematical model supported by architectural framework to enable the creation of formally confirmed security mechanisms for such systems. In this work, the authors are presenting architectural and implementational aspects to help IT experts in moving from the theoretical considerations to real system implementations. UML activity and component diagrams were used to illustrate a practical implementation of a data privacy mechanism for trans-domain systems.

Keywords: data privacy, e-services security, integration platforms, trans-domain systems.

Introduction

In numerous previous publications [10][12][13][14], the authors showed a lack of methodical (based on mathematical modelling and supported with architectural modelling) approach in designing and creating IT systems – particularly electronic platforms for public tasks handling. Because of many reasons like time pressure or budget limitations, design processes are mostly focused on architectural modelling which is often not supported by formally confirmed properties. Proper process is especially important when designing security mechanisms for such systems. Illegal breaches of data confidentiality may not only lead to loss of trust of users but some more serious problems as we are considering

Cite this Article as: Jaroslaw Wilk and Boleslaw Szafranski (2022)," Architectural and Implementation Aspects of Security Mechanisms for Digital Electronic Integration Platforms", Journal of Internet and e-Business Studies, Vol. 2022 (2022), Article ID 277290,
DOI: 10.5171/2022.277290

public electronic tasks (meaning that public administration security is affected).

The mathematical modelling part was presented by authors in previous publication [9] where detailed model was introduced. The authors also prepared a short paper [10] about how to integrate mathematical modelling with object-oriented diagrams (UML was selected). In this paper, the next step in the process is described. With the use of enterprise architecture, mathematical modelling is integrated with architectural modelling to present the design process of security mechanisms for public e-services integration platforms.

In this paper, the authors shortly reminded main concepts of their mathematical model (Chapter 2) so it can be used in architectural and implementational modelling. In Chapter 3, the authors discussed methodical guidelines for architectural modelling and suggested the process of building security mechanisms for trans-domain platforms in a prognostic approach. Next, in Chapter 4, five phases of the proposed process were presented as UML diagrams (with integration of mathematical elements). In Chapter 5, technological and implementational aspects of security mechanisms for digital electronic integration platforms was presented. Generalized component architecture of the trans-domain (responsible for integration to deliver complex e-services with components coming from different domains) platform was described with the clear indication of cooperation of security control system components with standard trans-domain platform elements. Components were presented with the use of UML with mathematical parts placed into them. Chapter 5 summarizes the considerations and presents next steps for the research in this field.

Mathematical Model – Main Concept

In this chapter, the main concept of mathematical model presented in details in [9] is reminded so it can be used in the next technological chapter (so all used symbols are explained before being used). The security control model for domain platforms (SM) consists of data and service security control models:

$$\langle P, D, Q, R, T, E, B, \rho, \tau, \delta, VF \rangle \quad (1)$$

where:

- P - the collection of entities $P = \{p_1, p_2, \dots, p_i, \dots, p_I\}$,
- D - the collection of data units $D = \{d_1, d_2, \dots, d_m, \dots, d_M\}$,
- Q - the collection of confidentiality classes $Q = \{q_1, q_2, \dots, q_h, \dots, q_H\}$,
- R - the collection of operations $R = \{r_1, r_2, \dots, r_n, \dots, r_N\}$,
- T - the collection of operations scopes $T = \{t_1, t_2, \dots, t_g, \dots, p_G\}$,
- E - the collection of services $E = \{e_1, e_2, \dots, e_l, \dots, e_L\}$,
- B - the collection of permissions categories $B = \{b_1, b_2, \dots, b_f, \dots, b_F\}$,
- ρ - flow relationship,
- τ - operation relationship,
- δ - service launch relationship,
- VF - the $\langle H_1, H_2, H_3, H_4, H_5 \rangle$ data security control functions and $\langle G_1, G_2 \rangle$ electronic services security control functions of the model (e.g., a function that does the flow verification or assigns confidentiality classes, scope of operations and categories of permissions).

The flow relation is built on the pairs of confidentiality classes: $\rho \subset Q \times Q$. The operation relation is built on pairs of operation scopes: $\tau \subset T \times T$. The service launch relation is built on pairs of permission categories: $\delta \subset B \times B$. When analyzing the relations of flows, operations and service launches, it can be concluded that in the case of meeting a partial order requirement in the sets of confidentiality classes, operation scopes and permission categories, it is possible to use the lattice theory [2]. The fulfilment of the partial order condition has its justification in the practically considered relations. Using this property, the flow (QL), the operation (TL) and the service launch (BL) lattices were defined:

$$QL = \left(Q, \rho, \oplus^{Q}, \otimes_{Q}\right) \quad (2)$$

where:

- Q is the partially arranged collection,
- $Q = \{q_1, q_2, \dots, q_h, \dots, q_H\}$,
- ρ is the relationship of the partial arrangement,
- $\oplus^{Q}$ the operator to set the supremum of its arguments,
- $\otimes_{Q}$ the operator to set the infimum of its arguments,

$$TL = (T, \tau, \oplus^{T}, \otimes_{T}) \quad (3)$$

$$BL = (B, \delta, \oplus^{B}, \otimes_{B}) \quad (4)$$

TL and BL lattices are created similarly to QL but with different arguments. Above lattices would formally describe the security rules for domain platforms and, if they were consistent, they would constitute the basis for constructing a super-lattice (IL) [8], reflecting common rules on the trans-domain (integration) platform:

$$IL = (Q \times T \times B, \leq, \oplus, \otimes) \quad (5)$$

It is important to mention that the presented model was created for public tasks integration platforms but can be used for any generic task also in almost any business (non-public platforms) environment – that's why previous "public tasks" are reflected as "defined tasks" in this article. The functionality of the electronic platform includes the possibility to handle defined tasks (Z) according to handling schemes (S) appropriate for particular tasks. The individual handling diagrams are clearly assigned to the individual defined tasks with the function π:

$$\pi: Z \rightarrow S \quad (6)$$

The defined task handling diagram defines a sequence of enforcements. Enforcements, which may occur in a given electronic platform, form the collection W. The sequence of enforcements, defined by the defined task handling scheme, S_k, define the handling process unambiguously:

$$w_1^k, w_2^k, \dots, w_n^k, \dots, w_{N_k}^k \quad (7)$$

Methodical guidelines for architectural modelling

Based on the authors' experience, analytical and construction works were conducted in accordance with the principles of enterprise architecture and with the use of object-oriented modelling. In such an approach, the architectural framework for creating security control mechanisms for electronic handling of public tasks using the effects of mathematical modelling will consist of two types of architectural frameworks:

- **functional architectural framework** (in short: functional framework) which is used to determine the structure and operating principles of security mechanisms based on the previously proposed lattice mathematical model [14]; the basic purpose of creating this type of framework is to express static and dynamic features of lattice-based model in the form of a set of UML diagrams as this is the language commonly used in designing IT solutions,

- **architectural management framework** (in short: management framework) constructed in order to determine a way of integrating the effects of mathematical modelling into the process of designing security mechanisms; thanks to such an approach, methodical conditions will be created for stimulating the process of designing security mechanisms with the results obtained from mathematical modelling. This means that mathematical modelling, especially lattice models will become a structural element of the architectural framework for the production of security mechanisms.

Taking the above into account, the authors reviewed and analysed UML diagrams in terms of assessing their suitability for constructing architectural frameworks of the two aforementioned types. Extensive results of this analysis and resulting recommendations can be found in [8]. Using these recommendations and the results of previous research [1][4][5][7][11], the authors assumed that:

- The functional framework is a collection of the following UML diagrams:
  - class and component diagrams - to reflect static elements of the model,
  - activity, use case, state and sequence diagrams - to reflect the dynamics of the model.

- The management framework is a set of activity diagrams:
  - reflecting the processes of execution of particular phases of the design process.

As functional framework with standard UML diagrams to reflect static and dynamic elements of the model is commonly used, authors focused in this paper on management framework which is often omitted by system architects (focusing on quick system implementation rather than on design and development process for the future).

Regardless of the approach used in the case of handling public tasks in a complex environment of electronic platforms operating in the model of a multilateral interoperability framework, i.e., with the intermediary role of a trans-domain platform, it is necessary to define common security rules for this environment (also known as a common security specification). The authors assumed that the way to reflect these common rules for the entire environment would be a super lattice taking into account (absorbing or including), under certain conditions, the rules existing in domain platforms. It is the super lattice that will be the basis for the functioning of the security mechanism operating on the trans-domain platform. Below, the processes of superclass construction for the trans-domain platform are considered, using the diagnostic approach in the first variant and the prognostic approach in the second variant. The use of a diagnostic approach takes into account the form of domain platform lattices (in general, these lattices do not change), which in turn leads to the creation of a super-lattice, which can be conventionally described as "weakly" integrated. The prognostic approach gives greater weight to the requirements that are formulated before starting the construction of the super-lattice and which result from the desired institutional and organizational rules. In this way, it is possible to obtain a structure that can be considered "strongly" integrated, because the form of local, domain-specific lattices directly results from the features of the super-lattice.

In order to construct a management framework for the **diagnostic approach**, the following phases of the process were distinguished:

1. **Identification** - extraction of features of security solutions for subsequent verification in the next step of compliance (non-contradiction) of security rules.
2. **Verification** - testing the inconsistency of domain security rules (applicable on domain platforms).
3. **Construction** – creation of resulting rules (aspect lattices and trans-domain super lattice) by taking into account specifics of domain rules, for the security mechanism of the trans-domain platform,
4. **Processing** - transformation of rules (lattices) of trans-domain security mechanisms, e.g., by consolidation of identical levels of security attributes,
5. **Evaluation** - verification, acceptance or rejection of the final security mechanism for the trans-domain platform and cooperating domain platforms.

For the **prognostic approach**, the phases of the process are as follows:

1. **Specification of assumptions and requirements** - the final form of the security requirements specification will be formulated iteratively as

follows. First, on the basis of the institutional and organisational rules (e.g., contained in the current security policy), a preliminary security requirements specification is created (most often by the chief security architect), which is forwarded to the domain platform managers in order to confront its provisions with the domain rules. The domain platform operators may submit, together with a justification, the need to modify the submitted specification in order to take into account the specificity of their security rules. If the arguments in the justification for change are accepted, appropriate modifications will be made to the security requirements specification. This procedure may be repeated until the final form of the security requirement specification is reached and approved.

2. **Construction** - Based on the approved specification, the aspect security rules are created in the form of aspect lattices for the trans-domain platform.
3. **Integration** - the assembly of the aspect security lattices is performed, resulting in a security super-lattice for the trans-domain platform.
4. **Processing** - the result of this phase is the creation of domain lattices, either directly based on the previously approved specification or by using the reduction operation of the trans-domain platform super lattice created in point 3.
5. **Evaluation** - in this phase there is a final verification of compliance of domain security rules, i.e., domain lattices with the rules of the trans-domain platform and with the super lattice of the trans-domain platform.

In accordance with the adopted assumption ("strong integration"), we will further focus on the approach conventionally called as prognostic.

Fig. 1 shows the process of constructing a security mechanism for trans-domain platforms based on the prognostic approach. The phases of the diagram are highlighted in green. Sub-processes performing particular tasks in subsequent phases are marked in blue.

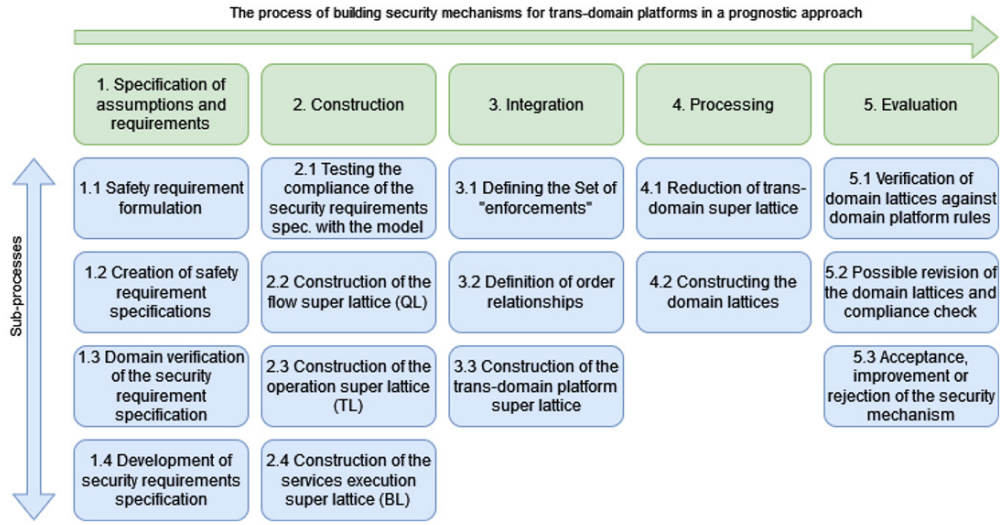

Fig. 1. Diagram of the process of constructing security mechanisms for trans-domain platforms in a prognostic approach

According to the previous findings and the results of the analysis presented in [10], the architectural management framework is a set of diagrams of the following UML notation activity diagrams:

- one diagram representing a comprehensive view of the process phases,
- five sub-process diagrams defining the structure and course of execution of each of the identified process phases.

Fig. 2 illustrates a diagram of the overall process of building a trans-domain platform security mechanism previously presented in block form in Fig. 1.

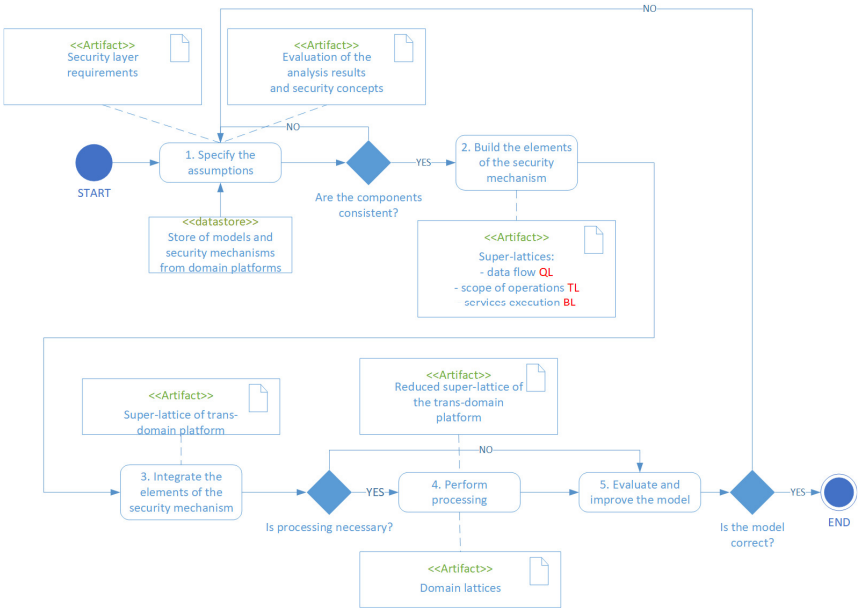

Fig. 2. Process of constructing security mechanisms for trans-domain platforms in a prognostic approach - activity diagram

The diagram (Fig. 2) shows the five phases of the process with the added artifacts used (institutional-organizational requirements and rules, security policies) and produced (security requirements specification, aspect lattices, domain lattices, trans-domain super lattice) in the process. The elements of the mathematical model have been added in red (lattices) and black (functions from [9]) to the UML diagrams presenting in details the process and sub-processes of each phase.

Architectural framework for phase 1 "specification of assumptions and requirements"

In the first phase, assumptions (specification of security requirements) for the security mechanism under construction are collected and developed. For this purpose, below inputs are used:

- the security requirements for the constructed and provided electronic services,
- already available security policies,
- indirectly (by assessing the compliance of the emerging security requirements specification with the domain specificity) the existing security models and mechanisms developed for domain platforms.

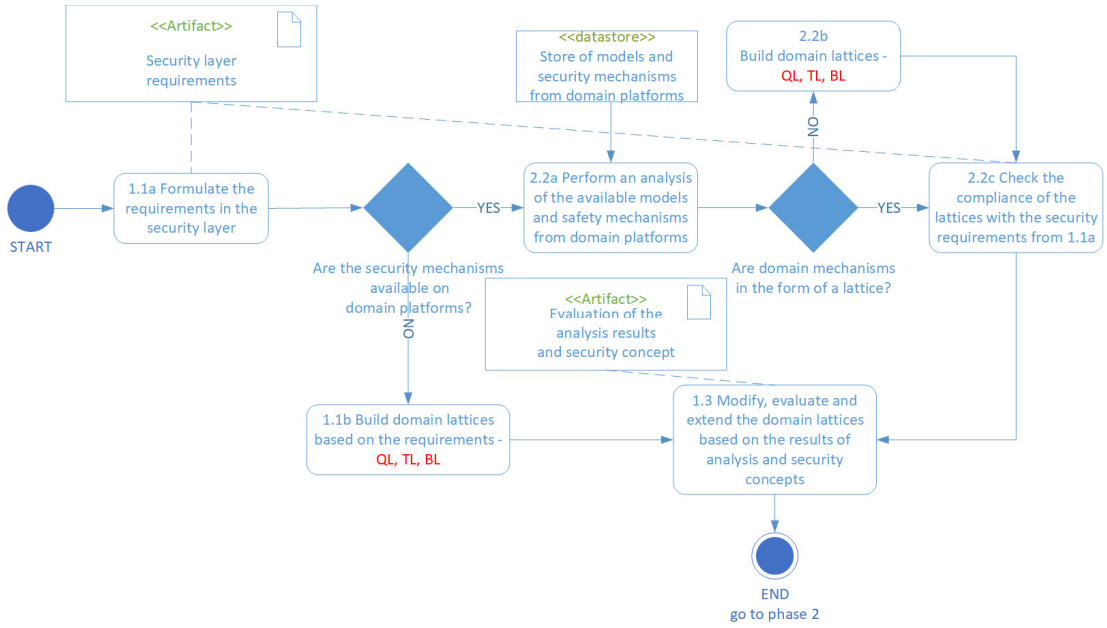

Fig. 3. Diagram of phase 1 "specification of assumptions and requirements" - activity diagram

Architectural framework for phase 2 - "constructing"

In the "construction" phase, basing on the specification of the security requirements from phase one, there is a transition to the aspectual forms of the QL data flow lattice, TL scope of operations lattice and BL service execution lattice. Before this can happen, however, the safety requirements are verified for their compatibility with the lattice model, i.e., whether the required partial order is present and whether the rules are inconsistent. In case of a negative verification result, a return to phase one is made to redefine the requirements

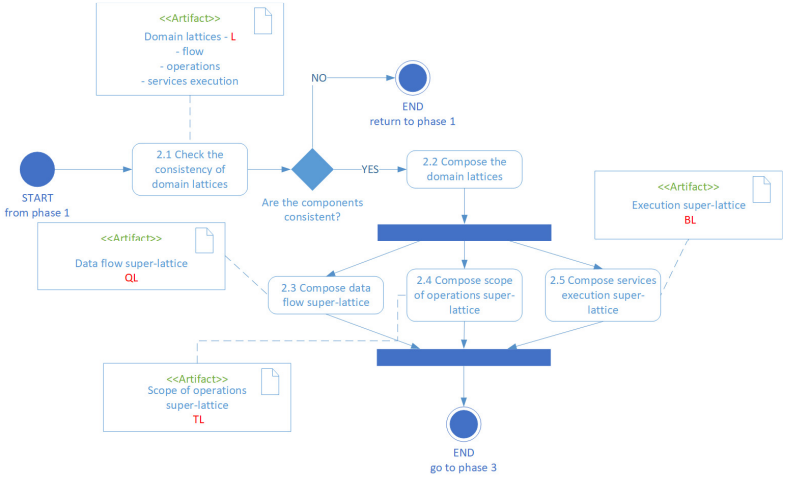

Fig. 4. Diagram of phase 2 "construction" - activity diagram

Architectural framework for phase 3 - "integration"

Phase three defines the process of assembling the aspect lattices into the super lattice of the trans-domain platform. The process starts with the definition of a set of constraints and the order relation between them (based on the aspect lattices) to finally construct a single super lattice covering the three aspects of security (data flow, scope of operations and service execution privileges). The explanation of mathematical functions (marked in black) and concept of an enforcements cube mentioned in the diagram is described in [9].

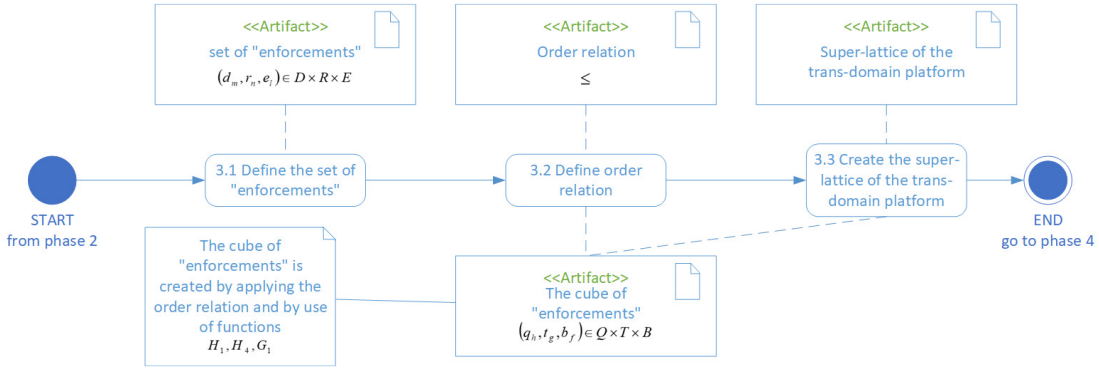

Fig. 5. Diagram of phase 3 "integration" - activity diagram

Architectural framework for phase 4 - "processing"

The fourth phase (shown in Fig. 6) implies the creation of security mechanisms for domain platforms in order to use them in the fifth phase (evaluation) and additionally to execute domain services there (without using the trans-domain platform, if necessary). It consists of additional processing of the developed security mechanism by reducing the superlattice of the trans-domain platform. The construction of domain super lattices is realised by taking into account the domain specificity and removing from the trans-domain lattice enforcement triples whose security attributes are not present in the considered domain platform (based on a pre-approved security requirements' specification).

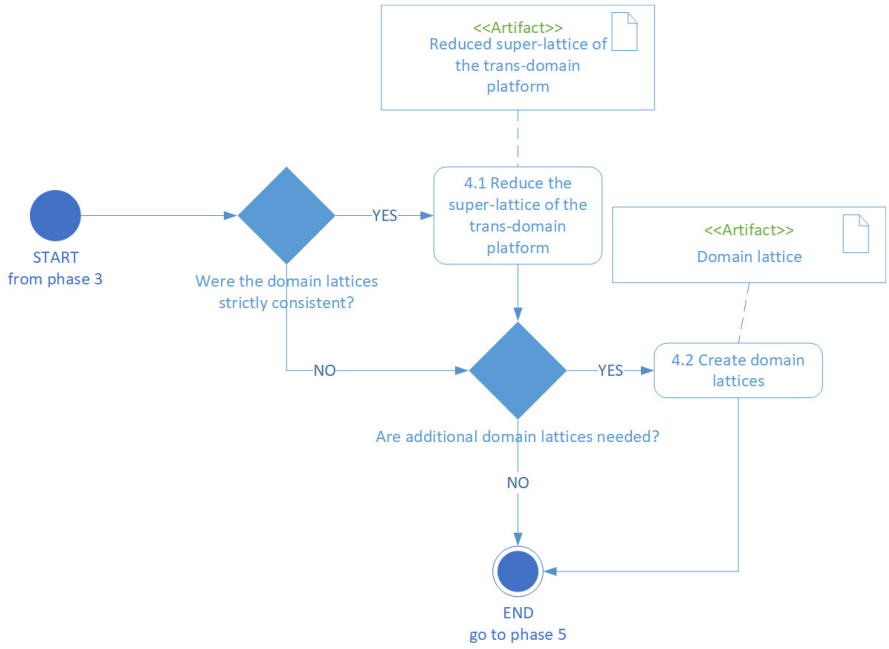

Fig. 6. Diagram of phase 4 'processing' - activity diagram

Architectural framework for phase 5 - "assessment"

In the last phase five (shown in Fig. 7), verification and improvement (if possible) of the built mechanism is carried out. In the first step, the compliance of the domain lattices (created from the trans-domain platform super lattice) with the security rules and policies of the domain platforms is checked. In case of non-compliance, this means that the security architect did not include some domain platform specific requirements in the prediction principals developed in phase one or they were included incorrectly. Depending on the relevance of the differences and the importance of the considered domain platform for the implementation of the trans-domain services, either the domain mechanisms will be improved (by extending the domain lattices or by returning to phase one to improve the developed security requirements specification) or the domain will be excluded from the scope of the developed security mechanism of the trans-domain platform. In the second step, the assembly of all checked and corrected domain lattices is performed in order to compare the obtained result with the super lattice of the trans-domain platform built in phase three. If the super lattices obtained are compatible, the process is completed. Otherwise, it is necessary to return to phase one in order to take into account the specific requirements of the domain platform that influenced the non-compliance.

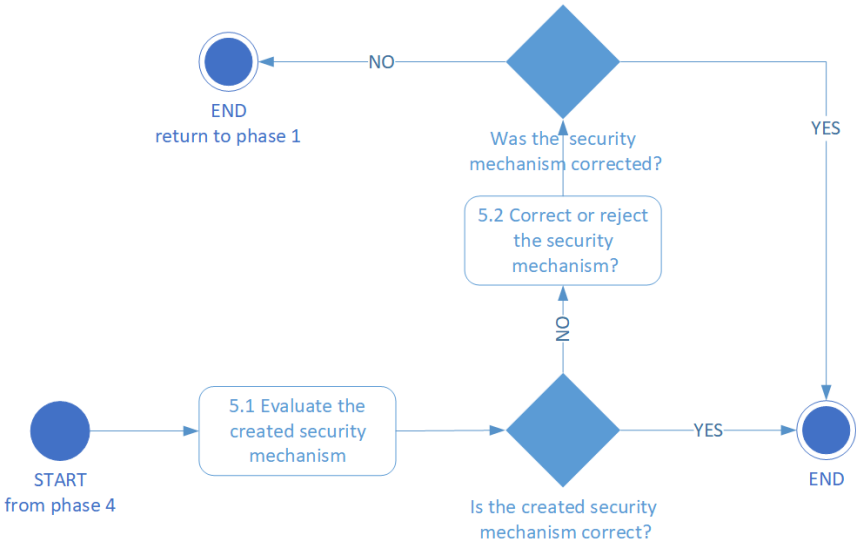

Fig. 7. Diagram of phase five "assessment" - activity diagram

Security mechanisms – technological and implementational aspects

Generalized component architecture of the integration platform

The developed architectural and management framework, taking into account the effects of mathematical and architectural modeling, can be practically used to design and produce security mechanisms of integration (trans-domain) platforms. Such mechanisms with formally confirmed properties make it possible to increase the security of automatic handling of defined tasks.

For the purposes of presenting the structural and implementation aspects of security mechanisms, the following components of a typical trans-domain platform can be distinguished:

- The presentation and access layer that most often provides access through a web portal along with an optional mechanism for handling other access channels, e.g., e-mail, access applications for mobile devices. It also provides mechanisms for communication with other systems, e.g., with the use of web services.
- The business logic layer consisting of the following systems: business logic and identity management, e-services management control, security management, content management and an integrating system, e.g., in the form of a communication bus, which will be able to handle fast and uninterrupted information flow between domain platforms and the trans-domain platform. The security mechanisms discussed in the article are part of the security system of the trans-domain platform, which is responsible for the secure implementation of all activities that also go beyond the verification of the defined task execution correctness (e.g., in terms of integrity, accountability, etc.).
- The data layer responsible for maintaining the catalogue of defined tasks, implementation schemes, sets of operations, data units, electronic services, entities and data supporting the operation of the system (including event log, addresses of domain platforms, additional content published on the access portal). From the

perspective of the article, the following are important:

- Authorization database - where the security attributes of entities, services and data containers are stored. In addition, the authorization database also retains the access rights to other elements of trans-domain platform that are not relevant from the perspective of this article, e.g., administrative rights to the platform itself and its components.
- Security model database – where lattice definitions for trans-domain security mechanisms are stored.

The specified elements are presented in the package diagram of the trans-domain platform (Fig. 8). Packages containing elements of the security mechanism for handling defined tasks (transformed to electronic services) are marked in blue.

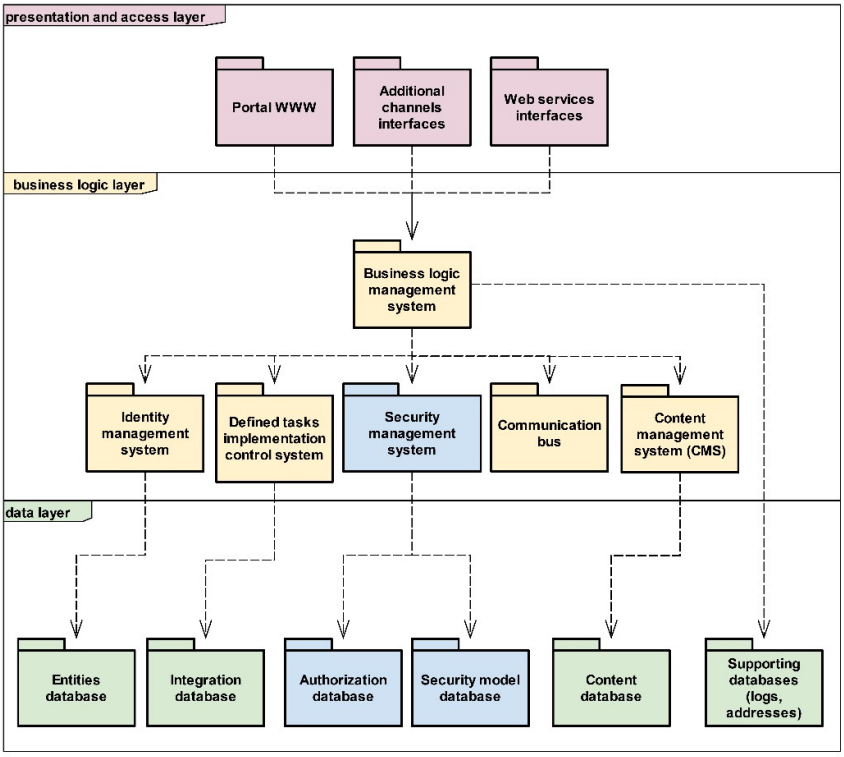

Fig. 8. Trans-domain platform package diagram

Fig. 9 shows a diagram of components containing elements necessary for the implementation of defined tasks (executed as e-services) with the use of trans-domain platforms. The security model is presented here as a single component of the lattice-based security mechanism (marked in blue). It is part of the security management system and is discussed in details in the next subchapter 4.2.

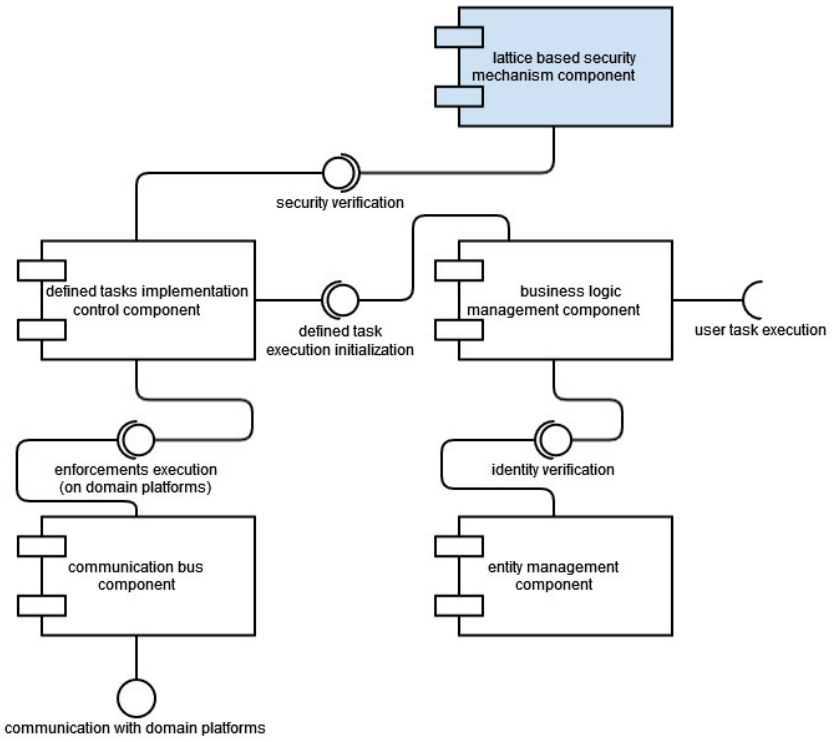

Fig. 9. Components diagram of the trans-domain platform with lattice-based security mechanism highlighted

As part of the detailed tasks execution, the business logic management component verifies the entity's identity in the entity management component. Then, the control is transferred to the defined tasks implementation control component, which, using a security lattice-based mechanism (marked in blue), verifies the authority of the entity (using the rules described in the mathematical model) to implement all the enforcements. In the event of a positive assessment, all required domain platforms are notified (via the communication bus component) about the admissibility of the execution of the enforcements.

Interaction between the security management system components and the components of the trans-domain platform

The security management system of the trans-domain platform consists of the components listed in Fig. 10, which are responsible for the management of (elements that fit in the lattice-based security mechanism presented in the article are highlighted in bold):

- **security attributes of entities (authorizations),**
- **security attributes of services and data units (confidentiality classes, operational scopes and permission categories),**
- **a developed security model and security verification (maintaining a security lattice model for trans-domain platforms),**

- security of communication with other electronic services platforms,
- other security attributes, including access, integrity, accountability and login.

Fig. 10 shows the components of the security management system of a trans-domain platform:

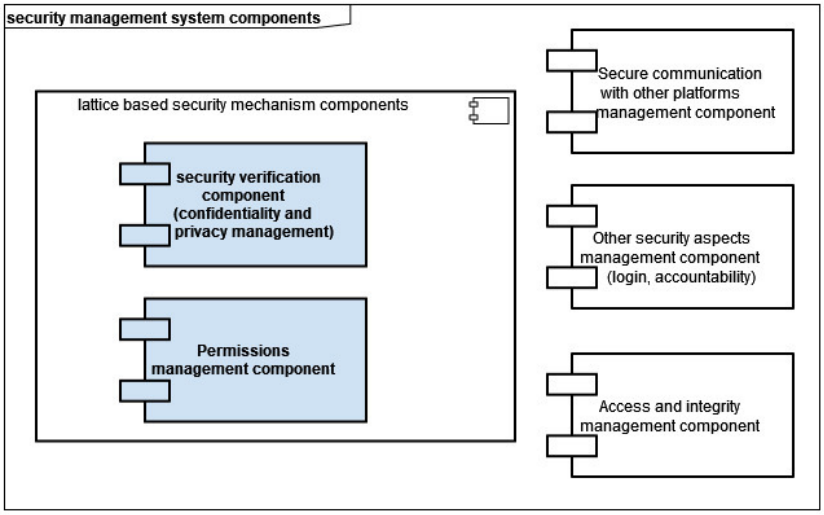

Fig. 10. Components of the security management system of a trans-domain platform.

The components of the lattice-based security mechanism (Fig. 10) were highlighted in blue: the security verification component (responsible for confidentiality management through the lattice-based security model) and the access rights management component. The other components were not considered in the article, they were listed for the sake of completeness of the description.

Fig. 11 is an extension of the diagrams presented in Figs. 8 and 9 with additional relations to the general components of a trans-domain platform.

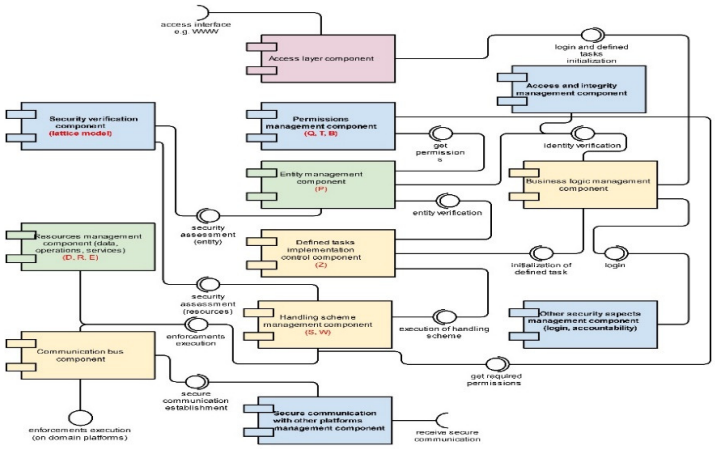

Fig. 11. Diagram linking the components of a security management system with general components of a trans-domain platform

The diagram (Fig. 11) retains the colors from Fig, 8, where the components from the presentation and access layer are marked in purple, the components from the business logic layer are marked in yellow, and the data layer marked in green. Additionally, the components included in the security management system are marked in blue and bold, and the elements of the lattice mathematical model are marked in red to illustrate the location of the lattice-based security mechanisms in the structure of the entire trans-domain platform.

Summary

The developed functional and management frameworks, taking into account the effects of mathematical and architectural modelling, can be practically used to design and produce security mechanisms of trans-domain platforms. Such mechanisms with formally confirmed properties allow increasing the security of automatic handling of public tasks. This paper, with previous publications [9][10], is an instruction of how to combine

mathematical and architectural modelling not only for trans-domain platforms but for any security components of integration systems design.

The next steps in this area could be the real implementation of the suggested solution based on currently available technology. There are many systems available on the IT market for building integration platforms including trans-domain platforms. The current list of leaders in the field of "Enterprise Integration Platform as a Service" was published by Gartner in 2020 [3] and it includes solutions from the following companies: Informatica, Boomi, SAP, Oracle, Workato, MuleSoft, Jitterbit, SnapLogic and Microsoft. Due to the dynamic growth of new systems built on cloud, the Azure Integration Services [6] from Microsoft were selected by authors for further considerations on the practical use of the developed model and architectural framework. Authors are also working on another related topic which is the efficiency evaluation of security mechanisms created on the basis of the lattice mathematical model. After first research, it was proven that although it is not much improving efficiency of trans-domain platforms during normal usage (services execution), we can gain significant efficiency improvement when extending the model with new enforcements (by adding additional electronic services or new domain platforms) [15].

Notes

“weakly” integrated is understood as having a less transparent, non-standardized integration, because it takes into account the specificity of security policies existing in independently developed separate domain platforms.

"strongly" integrated is understood as a reflection of the policy of top-down normalization, which enforces a close similarity or even identity of the security policies of the domain platforms interacting with each other through the trans-domain platform.

References

- Alghathbar K., Wijeskera D., Consistent and complete access control policies in use cases, In UML 2003 - The Unified Modelling Language, Model Languages and Applications, 6th International Conference, San Francisco, 2003, s.373–387.
- Denning D. E., Denning P. J., *Certification of Programs for Secure Information Flow*, Purdue University (1976).
- Gartner, Gartner Magic Quadrant for Enterprise Integration Platform as a Service, https://www.gartner.com, 2020 (date of access 03.2021).
- Jürjens J., Secure Systems Development with UML, Springer Verlag, 2004.
- Lodderstedt T., Basin D., Doser J., Secureuml, A UML-based modelling language for model-driven security, Proceedings of the International Conference on the Unified Modelling Language, UML’2002, 2002, s. 426–441.

- Microsoft, Azure Integration services, https://azure.microsoft.com (date of access 03.2021).
- Montangero C., Buchholtz M., Perrone L., Semprini S., For-lysa: UML for authentication analysis. In Global Computing, IST/FET International Work-shop, GC'2004, 2005, s. 93–106.
- Szafranski B., *Databases security processes modelling, with particular emphasis on their integration*, Military University of Technology, Warsaw (1987).
- Szafrański B., Wilk J., Mathematical modelling of processes to handle public tasks in the electronic platform environment, *Proceedings of the 36th International Business Information Management Association Conference (IBIMA)*, Granada, Spain (2020).
- Szafrański B., Wilk J., Integration of mathematical and object-oriented modelling in management of security mechanism designing for IT systems, *Proceedings of the 36th International Business Information Management Association Conference (IBIMA)*, Granada, Spain (2020).
- Talhi C., Mouheb D., Lima V., Debbabi M., Wang L., Pourzandi M., Usability of Security Specification Approaches for UML Design: A Survey, Journal of Object Technology, Volume 8, no. 6, ETH, Zürich, 2009.
- Wilk J., Electronic services security management for the public administration, *Computer Science and Mathematical Modelling No. 4*, *pages 25–32,* Military University of Technology in Warsaw, Poland (2016).
- Wilk J., Security of Composite Electronic Services, *International Journal of New Computer Architectures and their Applications (IJNCAA),* Vol. 5, No. 3, pages 127-140 (2015).
- Wilk J., Information security management model for integration platforms, *e-Technologies and Networks for Development (ICeND) – IEEE Explore*, The Fourth International Conference on e-Technologies and Networks for Development, Lodz, Poland (2015).
- Wilk J., *Efficiency of lattice-based security mechanisms supporting public tasks on digital integration platforms*, Computer Science and Mathematical Modelling No. 13, (in print).