2021 16th International Conference of System of Systems Engineering (SoSE) | 978-1-6654-4454-5/21/$31.00 ©2021 IEEE | DOI: 10.1109/SOSE52739.2021.9497461

Architecting System of Systems Solutions with Security and Data-Protection Principles

Behnam Asadi Khashooei*, Alexandr Vasenev*, Hasan Alper Kocademir†, Roland Mathijssen*
* Embedded Systems Innovation (ESI), Netherlands Organisation
for Applied Scientific Research (TNO), Eindhoven, The Netherlands
Emails: {behnam.asadikhashooei, alexandr.vasenev, roland.mathijssen}@tno.nl
†Roche Diagnostics International AG, 6343 Rotkreuz, Switzerland
Email: alper.kocademir@contractors.roche.com

***Abstract*—The rapid advancement of communication technology realized the dream of interconnected systems. In addition to enabling scalability and flexibility of solutions, this paradigm created new system design challenges. One such challenge is to holistically address security and privacy concerns of solutions early in design while respecting the system of systems context. This paper proposes a method for the concept design phase on how to create design alternatives with the help of security and data-protection principles. The outcome is a set of design concepts that reflect stakeholders' concerns and best practices.**

***Index Terms*—security engineering, system architecting, system of systems**

I. Introduction

Recent years showed a rapid increase of continuously interacting systems, making system of systems (SoS) concept today's reality. More and more often, modern solutions that bring value to customers should rely on existing SoS parts. Next to this is the growing awareness that systems almost always interact with their environment, that consists of other systems. And as systems begin interacting emergent behavior will become more important.

Clearly, novel solutions must pay attention to aspects like cybersecurity and privacy. For manufacturers, it means ensuring market access by fulfilling regulations, e.g., EU GDPR for privacy [1], or IoT Security Law of California [2] for securing connected devices. Moreover, system security supports trust in the manufacturer's brand, which is an important differentiator between companies. A weak focus on security leads to flawed, unsustainable, and very costly solutions due to recalls or fixes in the field. A system ill-designed for security endangers the customers' trust and empties the manufacturer's wallet, it can even put the safety of a population at risk. Customers are exposed to issues like data leaks, under-performing, or unavailable systems. Development will bear extra costs when integrating the system into a customer's secure network, or by missing out on early focus on certification and compliance. Moreover, a security-unaware architecture hampers the opportunities to introduce new features for service of the systems, e.g., remote predictive maintenance and limits the gathering of field knowledge for potential improvements in systems design [3].

Since SoS consists of independent and interdependent systems, creating a solution that utilizes SoS elements is complex. As pointed out by [4], these constituent systems are often coming together in ways that they may not have originally been designed for. Yet, designing for secure solutions means starting by considering security upfront, e.g., at the concept generation stage. Having suitable methods to create design options to meaningfully choose from is desirable. Furthermore, system creators should strike a golden balance between security and other system qualities to address business needs without expensive over-doing [5]. This requires that the attention for security has to be on a similar level as the other system qualities. Neglecting allocations of security measures (like anti-viruses or encryption) leads to wrong trade-offs between security, performance, costs, usability, and possibly other system qualities. This also requires the adaptation of current methods for systems architecting, e.g., model-based system engineering to allow considering security and privacy [6], [7]. On the other hand, the creation of tailor-made methods that combine the experiences of security engineering e.g. threat modeling [8], and system architecting methodologies e.g. CAFCR methodology [9] is essential. This paper is an effort in the latter direction.

This article argues that applying security and data-protection principles [1], [10] can assist in generating security-aware and privacy-aware solution concepts within a system of systems. This is because principles reflect best practices and are easy to communicate to stakeholders. In particular, we outline and illustrate a method for a security-aware authentication and data transfer solution that utilizes several SoS elements.

The proposed method aims to assist solution/system architects in balancing of key-drivers in system design. This design method is based on security and data-protection principles and encourages the adoption of security-by-design and privacy-by-design philosophy during the concept phase of a system design [11]. Furthermore, using this method provides the architect with a holistic view of the system and creates traceability from the high-level requirements in terms of key-drivers to implementations in terms of technical architecture.

This research is carried out as part of SECREDAS and INTERSECT projects under the responsibility of ESI (TNO). SECREDAS is funded by the ECSEL Joint Undertaking of the European Union under grant agreement number 783119. INTERSECT is a public private partnership funded by the Dutch National Research Council (Grant NWA.1162.18.301).

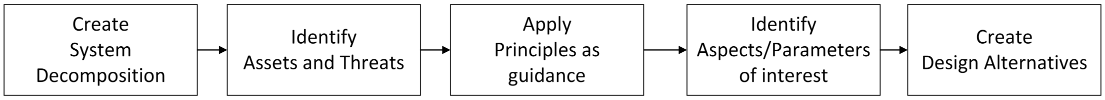
Create System Decomposition
Identify Assets and Threats
Apply Principles as guidance
Identify Aspects/Parameters of interest
Create Design Alternatives

Fig. 1. Proposed design method.

The proposed method can be summarized in 5 steps as shown in Figure 1. These steps are iterative, meaning one can go back and forth as the design matures.

This paper is structured as follows: Section II provides background on systems of systems challenges and opportunities. It also provides preliminary material used in the following sections. Section III contains the proposed method for developing design alternatives. Section IV contains the illustrative use case example for the proposed method and, finally, Section V provides a short discussion and concludes the paper.

II. Background and Related Work

A. Challenge and enablers for designing solutions for SoS

Architecting a solution for an SoS is challenging due to extensive interconnections and the expectations to rely on parts of the SoS already in place. Earlier assumptions and implementations may impose limitations to new solutions, for instance, because of existing communication channels and functionalities. Constrains and complexities call for creative thinking by architects who shall address multiple dilemmas.

Solution architects who develop a solution within the SoS context can particularly benefit from having:

1) An overview of relevant system, customer, and business aspects.
2) A guiding structure to generate alternatives, guide discussions with stakeholders, and make justified choices.

An overview of the larger context can inform developers about other relevant SoS elements and aspects. It includes stakeholders, functionalities, and communications [12]. It can also take into account different values in social contexts and their possible conflicts (see, e.g., [13] and the references therein). Such an overview can be formulated in customer, technical, and business terms. This set of elements is common for reference architectures and facilitates the decision-making for architects [14], [15]. This view is in line with the spirit of the CAFCR framework [9] as it facilitates a holistic view from the customer to the realization within product development.

Design principles provide a solid base for guiding system developers. As noted in [16], the success of using architecting principles (stable intermediate forms, policy triage, leverage at the interfaces, ensuring cooperation) correlates with the success of system development for SoS. Such condensed and somewhat abstract know-how guides decisions and helps to prevent the re-occurrence of design flaws. One example is the set of TRIZ (theory of the resolution of invention-related tasks) [17] principles (segmentation, taking out, asymmetry, and others) used for developing new systems. Another example is security principles by [18], such as open design and separation of privileges to ensure building secure systems from early on. The importance of having principles for guidance can be highlighted by including them into reference architectures [19].

Guidance to system developers is commonly formulated as methods that use principles. TRIZ utilizes principles to solve dilemmas. Methods for designing privacy-aware high-tech systems build on GDPR (General Data Protection Regulation) principles and PbD (Privacy-by-Design) [20]. Security methodologies apply security principles. For instance, [21] describes how the idea of security-by-design led to relating attacks to use cases, while the defense-in-depth principle causes limiting personal data access to doctors, but not system administrators.

The use of security principles together with guidance documents can help architects to generate security-aware solution concepts within a system of systems. These concepts can include different SoS elements and thus benefit from existing functionalities, data flows, and communications within SoS.

Before outlining and illustrating the approach, the paper briefly outlines structures of security and data-protection principles.

B. Security and data-protection principles

Security and data-protection principles are condensed abstractions of best practices that reflect domain expertise. Principles help designers and architects to discuss the rationale, inter-relations, trade-offs, and preferences. They inform the decision-making.

1) Data-protection principles: Table I shows GDPR principles [1], which are binding for EU and other countries processing EU citizens' data. GDPR principles build on previous efforts to ensure the privacy of individuals and aggregate earlier data-protection principles, e.g., from OECD [22].

TABLE I
List of GDPR principles

Numbering	Data-protection Principle
A	Lawfulness, fairness and transparency
B	Purpose limitation
C	Data minimization
D	Accuracy
E	Storage limitation
F	Integrity and Confidentiality
G	Accountability

2) Security principles: The security principles have been conducted from experience and provide useful guidance that contributes to a design with fewer security flaws. Table II summarizes the 9 security design principles that were suggested in seminal work of [18]. Since then many adaptations of the principles have been introduced (e.g. see [23]), however,

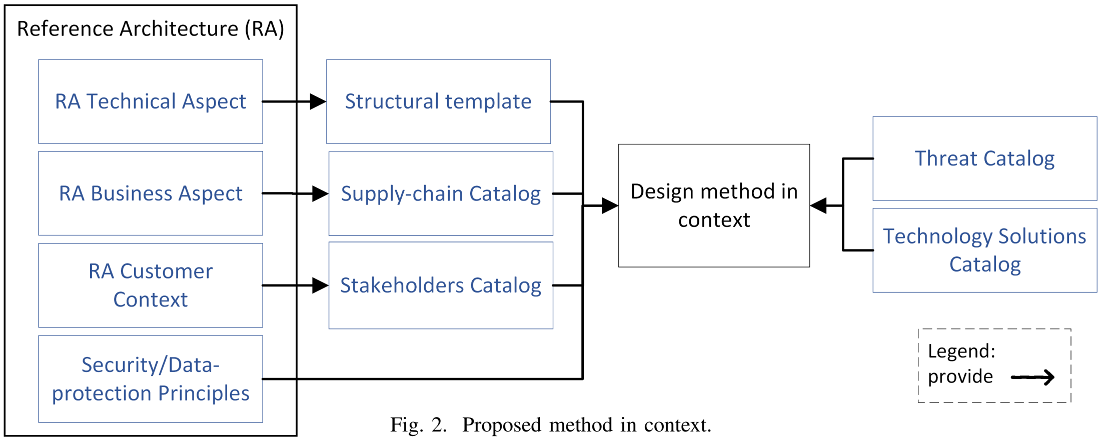

Fig. 2. Proposed method in context.

for our purpose, the list in [18] provides a sufficient level of abstraction.

TABLE II
LIST OF SALTZER/SCHROEDER SECURITY PRINCIPLES

Numbering	Security Principle
A	Least privilege
B	Least common mechanism
C	Open design
D	Economy of mechanism/ Keep it simple
E	Fail-safe defaults
F	Complete Mediation
G	Separation of Privilege
H	Psychological Acceptability
I	Compromise recording

Note that although the headlines of the principles in Table I and Table II are chosen to ease communication, applying them requires the knowledge of their full description by all parties involved in the decision-making process. Furthermore, one can choose a different set of principles than those mentioned in this paper which can also reflect the essence of certain standards.

So far, we have provided background on challenges and enablers for security and privacy-based design in SoS context. As we mentioned earlier, creating a solution that utilizes SoS elements is complex. However, this complexity can be managed by considering security upfront, e.g., at the concept creating stage. Therefore, one can benefit from simple yet flexible methods to create design options that facilitate the introduction of these security and privacy concerns into the system development life-cycle.

In the next section, we will introduce a method that enables the solution architect in the early phases of concept development while using the principles as a guide.

III. PROPOSED METHOD

We first start with the context in which the method is developed as illustrated in Figure 2, which summarizes the assumptions in using the proposed method. The first assumption is that the Technical, Business, and Customer contexts of the SoS are known (e.g., described in a relevant reference architecture). Secondly, the relevant security and data-protection principles are already identified and a general threat catalog is available. The latter can be in a form of a generic list of information security threats common for this type of systems. Furthermore, in Figure 3 we illustrate, how the listed elements of context are utilized in our proposed method.

In what follows, we provide a detailed description of each step in the proposed method.

A. Create System Decomposition

As a first step, the architect creates a high-level system decomposition by using the structural template from the technical SoS aspect. The main subsystems are identified to ensure a holistic view of the design structure.

B. Identify Assets and Threats

After creating the system decomposition, the architect identifies the main assets of the solution. A method like Crown-Jewels [24] to identify assets can be useful. Afterward, potential threats to these assets can be identified, e.g., with a threat catalog (for examples of threat catalogs see [25], [26]). The identified threat from the catalog can be further elaborated to account for the specifics of the design at hand.

As mentioned, this method is iterative, and therefore more threats can be identified as the design is evolving.

C. Apply Principles at guidance

To counter the identified threats, the architect can use principles (Tables I and II) as a way to consider mitigation solutions. As an example, the principle of complete mediation in Table II suggests that each access to any assets should be validated. Given that the assets and their corresponding threats are already identified, one can now use this principle as a guide towards solutions to mitigate the threat for unauthorized access to sensitive data.

D. Identify Aspects/Parameters of interest

In this step, the architect identifies aspects helpful to compare alternative solutions. These can originate from the investigation of users of the system, the owners of the business that are acting as part of the supply-chain for the created solution, the use case definition, etc. Such information can be obtained from a suitable reference architecture in the form of stakeholder and supply chain catalogs as depicted in Figure 2 and Figure 3. Furthermore, Key Performance Indicators (KPI)

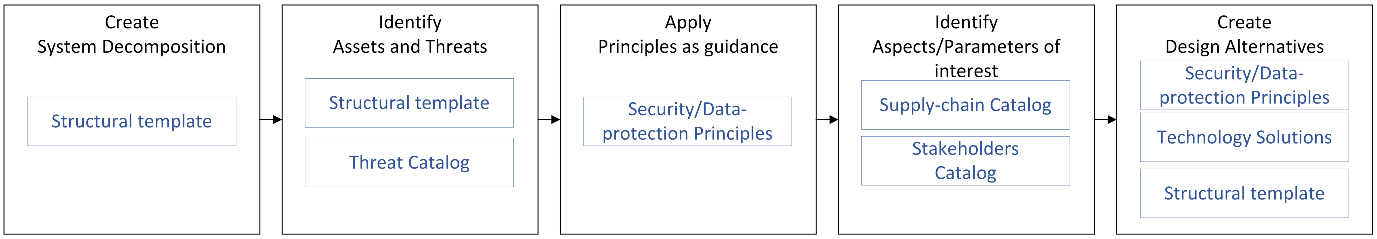

Fig. 3. Proposed method utilizing elements from context in Figure 2.

of the design are identified at this step. These design KPIs are connected with the stakeholders' concern and can help to evaluate designs. The residual risk to identified threats and the integration cost are examples of these design KPIs (see e.g. [27] and the references therein).

E. Create Design Alternatives

The final step is to create design alternatives by selecting technology solutions that provide the required functionality, as well as address the threats. A technical solution catalog can be used as a structural template to employ well-known design patterns or technologies. Furthermore, the mentioned principles can guide the designer in mitigating security and privacy concerns.

In this step, multiple design alternatives can be generated and assessed with the chosen aspects of interest. The outcomes can be discussed with other stakeholders to select an appropriate design. The selection of a final solution is subject to the trade-offs that naturally come with multiple-candidate solutions.

In the following section, we illustrate the method with the help of a use case from SECREDAS project [28].

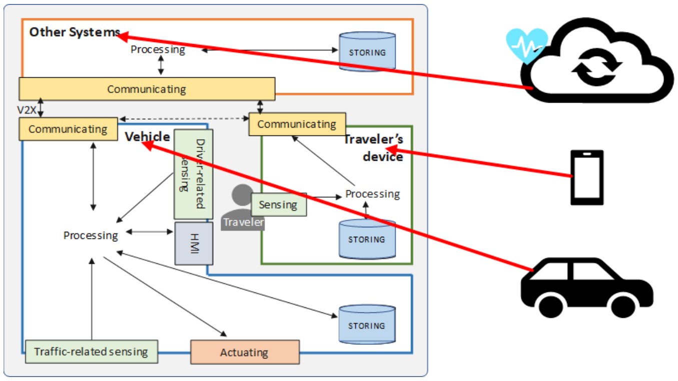

Fig. 4. System Decomposition and mapping to a technical reference architecture. This reference architecture integrates the domains of automotive and healthcare for SECREDAS project [28] and as such, the architecture contains elements from these domains and integrates them in a holistic view.

IV. ILLUSTRATION: HEALTHCARE DATA EXCHANGE

This use case focuses on the validation of a human driver's fitness to prevent an individual with certain health conditions to drive a vehicle. In particular, the driver's healthcare information is delivered from the healthcare information cloud to the car he wants to drive. Within the car, a fit-to-drive application validates the fitness of the driver.

This use case lies in the intersection of two SoSs: the healthcare information cloud and the vehicle. It also provides a service that needs to ensure the secure transmission of data while complying with the privacy regulations.

In the illustrative example, we focus only on part of the use case that corresponds to providing the data from the healthcare information cloud to the car. Furthermore, we omit details of the full design available in [28] and instead, provide a simplified overview of the actual design to illustrate the use of the proposed method.

A. Create System Decomposition

The first step is to create a system decomposition for the problem at hand. In this case, the architect chose to decompose his design. Consider 3 subsystems (as a subset of a technical reference architecture outlined in [29] as shown in Figure 4. The system of interest consists of a healthcare information cloud, a user device, and a car. Note that these entities can already exist as SoS parts.

B. Identify Asset and Threats

The main asset in this use case is the healthcare data. Such highly private sensitive data are subject to the GDPR and need to be protected against unauthorized access and processing. For this example, the architect identifies potential threats by using a threat catalog of the top-10 list of threats in automotive applications [26], [28] summarized in Table III. The architect focuses on Threat VIII "Attacks on privacy or data lost and leakage" and elaborates further with threats in Table IV.

TABLE III
TOP-10 AUTOMOTIVE SECURITY AND PRIVACY THREATS [28]

Numbering	Threat description
I	Attacks on backend server.
II	Attacking a car using V2X communication channels.
III	Attacking a car by exploiting software update.
IV	Social engineering or exploits vulnerabilities and weaknesses introduced by human errors.
V	Attacking a car's interfaces and functions for external connectivity.
VI	Attacks on in-vehicle network or software of on-board systems.
VII	Attacks that exploit security flaws in system design.
VIII	Attacks on privacy or data lost and leakage.
IX	Physical manipulation of on-board systems to enable an attack.
X	Attacks on sensors.

TABLE IV
ELABORATED THREAT-LIST FOR ILLUSTRATIVE EXAMPLE [28]

Numbering	Threat description
1	Attacker tries to login to Healthcare Information Cloud as if it was a legitimate user.
2	Attacker tries to listen in on the established connection between user device and Healthcare Information Cloud.
3	Attacker steals the user device where initial authentication to Healthcare Information Cloud has been done and tries to access up-to-date health data received from Healthcare Information Cloud.

C. *Apply Principles as guidance*

To address the threats, the privacy and security principles from Table I and Table II can be used. The architect realizes that the principle of integrity and confidentiality highlights the importance of assurance of data confidentiality in the communication channel. It guides her to consider (potentially, by discussing with a security specialist) how to protect the data in the communication channel between car and phone. Furthermore, the principle of complete mediation entails that every access to any assets is validated. It guides her to select a mechanism that ensures that the health data is not accessed without authorization.

D. *Identify Aspects/Parameters of interest*

One of the solution stakeholders is the driver whose personal data is at stake. Apart from that, the car manufacturer, as well as the cloud service providers, are important parties that could constrain the design in terms of available solutions. The attacker is another important stakeholder, although his actions are of an adversarial type. Following this thread of reasoning, the architect identifies a number of design KPIs e.g. the cost indication, the protection level against identifies threats, and the integration effort of each design alternative.

E. *Create Design alternatives*

The architect considers how data are transferred from the healthcare information cloud to the car through a user device. Given the identified threats 1 and 3 from Table IV and the complete mediation principle, one needs to ensure that health data are only accessed by an authorized person. The architect chooses to implement a multi-factor authentication solution [30] that uses a face recognition technique. In this way, a camera is needed to acquire the image of the driver for authentication. Here two alternative solutions can be identified: to use a user device's camera, e.g. a phone; or a camera available in the car.

Furthermore, to address threat 2 from Table IV and using the principle of integrity and confidentiality, the architect concludes that the communication channel between the user device and the healthcare information cloud must be protected. The architect chooses a well-established technology solution of Transport-layer-security (TLS) [31] in this communication channel. An alternative design can be to enhance the communication channel security by using Virtual Private Network (VPN) [32] together with TLS for the communication channel between the user device and the car.

These considerations lead to four different designs as depicted in Figure 5. Furthermore, Figure 6 shows how these designs reflect the principles applied when creating the design options.

Finally, in order to be able to compare and evaluate these designs in the decision-making process, the architect fills out the information regarding the aspects of interest from step D for each design alternative. Table V provides an example of such filled information in a table format.

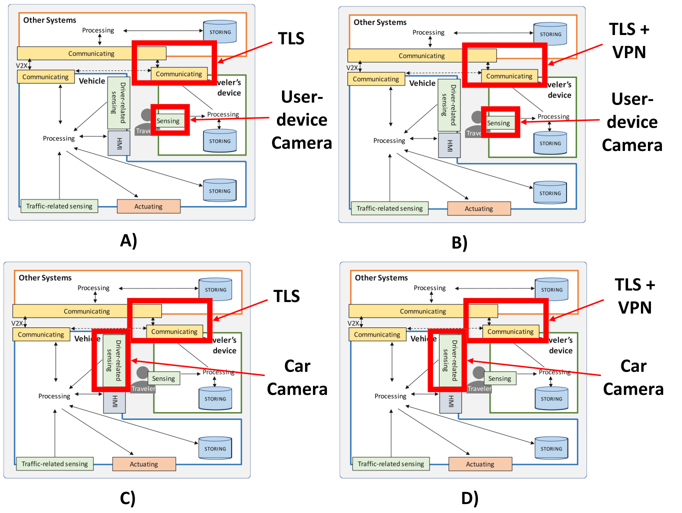

Fig. 5. Four design alternatives that were developed for the illustrative use case. Designs A and C use TLS in the communication channel between the car and user device while design B and D use TLS + VPN in this communication channel. On the other hand, Designs A and B use the user's device for authentication while Designs C and D use the car camera.

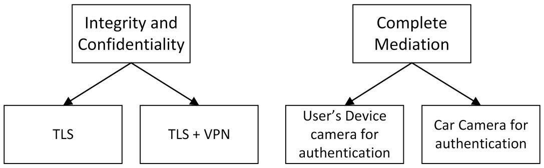
Integrity and Confidentiality
Complete Mediation
TLS
TLS + VPN
User's Device camera for authentication
Car Camera for authentication

Fig. 6. Tracing the principles in creation of design alternatives.

TABLE V
EXAMPLE OF FILLED-IN TABLE WITH DESIGN ALTERNATIVES DATA.

Aspects of interest	Design A	Design B	Design C	Design D
Component Cost	Low	Medium	Medium	High
Protection level for threat 1	High	High	High	High
Protection level for threat 2	Medium	High	Medium	High
Protection level for threat 3	Medium	Medium	High	High
Integration effort (1-10)	3	5	6	8
Tiers involved	1, 2	1,2	OEM, 1, 2	OEM, 1, 2

V. DISCUSSION AND CONCLUSION

The combined information in Table V, Figure 5 and Figure 6 can be depicted in a one-pager which will ease the communications and can be used in the final decision-making involving other stakeholders by creating trade-offs for different design alternatives. Furthermore, Table V resembles the well-known Pugh matrix [33] and can be further elaborated by filling it out with numbers and introducing weighting factors for the aspects of interest. It should be noted, that in the design of SoS other system qualities like performance, availability, etc. can be of crucial importance. Our proposed method can also incorporate these aspects in a similar fashion as Table V. For example, in the case of performance, the resource utilization of the alternatives can be evaluated, or when availability is important, the redundancy of designs can be assessed. Finally, we would like to emphasize that although this method has an intuitive style, it requires extended knowledge and experience both in security/privacy aspects and system architecting for its application in SoS.

This paper proposed a method for early design phase that generates design concepts for solutions within the SoS context. The design outcome is a set of holistic system views, where security and privacy concerns are linked to system building blocks and main information about alternatives solutions can be presented in a simple intuitive way. Furthermore, we provide an illustrative use case from the SECREDAS project in which we applied the proposed method to provide a set of alternative design solutions[1].

For future work, we plan to carry out further research on the use of this method in practice and to further extend it to the requirements engineering domain by focusing on the use of principles as a guidance.

References

[1] G. D. P. Regulation, "Regulation eu 2016/679 of the european parliament and of the council of 27 april 2016." Official Journal of the European Union. Available at: http://ec.europa.eu/justice/data-protection/reform/files/regulation_oj_en.pdf, 2016. Online; accessed 28 January 2021.

[2] L. C. Digest, "Senate bill no. 327, chapter 886. an act to add title 1.81.26 (commencing with section 1798.91.04) to. part 4 of division 3 of the civil code, relating to information privacy.." Approved by Governor September 28, 21 2018. Filed with Secretary of State September 28, 2018. Available at: https://leginfo.legislature.ca.gov/faces/billTextClient.xhtml?bill_id=201720180SB327, 2018. Online; accessed 29 January 2021.

[3] E. Sperling, "IIoT comes to chip manufacturing." Applied Materials; Available at:https://www.appliedmaterials.com/nanochip/nanochip-fab-solutions/september-2015/ed-sperling, 2015. Online; accessed on 2 February 2021.

[4] D. Ki-Aries, S. Faily, H. Dogan, and C. Williams, "System of systems characterisation assisting security risk assessment," in *2018 13th Annual Conference on System of Systems Engineering (SoSE)*, pp. 485–492, IEEE, 2018.

[5] H. G. Goldman, "Building secure, resilient architectures for cyber mission assurance," *The MITRE Corporation*, 2010.

[6] D. Mazeika and R. Butleris, "Identifying security issues with mbse while rebuilding legacy software systems," in *2020 IEEE 15th International Conference of System of Systems Engineering (SoSE)*, pp. 83–86, IEEE, 2020.

[7] D. Mažeika and R. Butleris, "Integrating security requirements engineering into mbse: Profile and guidelines," *Security and Communication Networks*, vol. 2020, 2020.

[8] A. Shostack, *Threat modeling: Designing for security*. John Wiley & Sons, 2014.

[9] G. Muller, "CAFCR: A multi-view method for embedded systems architecting," *Balancing Genericity and Specificity*, 2004.

[10] C. C. Lamb, "A survey of secure architectural principles," tech. rep., Sandia National Lab.(SNL-NM), Albuquerque, NM (United States); Sandia ..., 2015.

[11] K. Forsberg and H. Mooz, "The relationship of system engineering to the project cycle," in *INCOSE International Symposium*, vol. 1, pp. 57–65, Wiley Online Library, 1991.

[12] G. Muller, "Are stakeholders in the constituent systems sos aware? reflecting on the current status in multiple domains," in *2016 11th System of Systems Engineering Conference (SoSE)*, pp. 1–5, IEEE, 2016.

[13] J. Van den Hoven, P. E. Vermaas, and I. Van de Poel, *Handbook of ethics, values, and technological design: Sources, theory, values and application domains*. Springer, 2015.

[14] R. Cloutier, G. Muller, D. Verma, R. Nilchiani, E. Hole, and M. Bone, "The concept of reference architectures," *Systems Engineering*, vol. 13, no. 1, pp. 14–27, 2010.

[15] B. van der Sanden and A. Vasenev, "Architectural guidance in automotive for privacy and security: survey and classification," in *2020 IEEE International Systems Conference (SysCon)*, pp. 1–8, IEEE, 2020.

[16] M. W. Maier, "Architecting principles for systems-of-systems," *Systems Engineering: The Journal of the International Council on Systems Engineering*, vol. 1, no. 4, pp. 267–284, 1998.

[17] Y. Salamatov, *TRIZ: the right solution at the right time: a guide to innovative problem solving*. Insytec B.V., The Netherlands., 1999.

[18] J. H. Saltzer and M. D. Schroeder, "The protection of information in computer systems," *Proceedings of the IEEE*, vol. 63, no. 9, pp. 1278–1308, 1975.

[19] US Department of Defence, "US department of defence (DoD) reference architecture description."

[20] G. M. Riva, A. Vasenev, and N. Zannone, "Sok: Engineering privacy-aware high-tech systems," in *Proceedings of the 15th International Conference on Availability, Reliability and Security*, pp. 1–10, 2020.

[21] E. B. Fernandez and M. M. Larrondo-Petrie, "A methodology to develop secure systems using patterns," in *Information Security and Ethics: Concepts, Methodologies, Tools, and Applications*, pp. 654–670, IGI Global, 2008.

[22] OECD, *OECD Guidelines on the Protection of Privacy and Transborder Flows of Personal Data*. OECD Publishing, 2002.

[23] T. V. Benzel, C. E. Irvine, T. E. Levin, G. Bhaskara, T. D. Nguyen, and P. C. Clark, "Design principles for security," tech. rep., Department of Computer Science, GSOIS, Naval Postgraduate School, Monterey, CA, 2005.

[24] L. S. Metzger, G. Rebovich Jr, R. A. Cormier, S. J. T. Norman, D. L. Schuh, P. A. Smyton, R. S. Swarz, and F. C. Wendt, "Systems engineering guide: Collected wisdom from mitres systems engineering experts," tech. rep., MITRE CORP BEDFORD MA BEDFORD United States, 2014.

[25] R. Ross, M. McEvilley, and J. Oren, "Systems security engineering: Considerations for a multidisciplinary approach in the engineering of trustworthy secure systems," tech. rep., National Institute of Standards and Technology, 2016.

[26] W. UNECE, "Grva,"draft recommendation on cyber security of the task force on cyber security and over-the-air issues of unece wp. 29 grva."," 29.

[27] T. Haponava and S. H. Al-Jibouri, "Identifying the kpis for the design stage based on the main design sub-processes," in *Proceedings of joint CIB conference on Performance and Knowledge Management. June 3-4, Helsinki, Finland*, pp. 14–23, CIB, 2008.

[28] "Secredas project: an ecsel joint undertaking." https://secredas-project.eu/, 2021. Online, accessed 27 january 2021.

[29] N. Marko, A. Vasenev, and C. Striecks, "Collecting and classifying security and privacy design patterns for connected vehicles: Secredas approach," in *International Conference on Computer Safety, Reliability, and Security*, pp. 36–53, Springer, 2020.

[30] A. Ometov, S. Bezzateev, N. Mäkitalo, S. Andreev, T. Mikkonen, and Y. Koucheryavy, "Multi-factor authentication: A survey," *Cryptography*, vol. 2, no. 1, 2018.

[31] S. Turner, "Transport layer security," *IEEE Internet Computing*, vol. 18, no. 6, pp. 60–63, 2014.

[32] R. Venkateswaran, "Virtual private networks," *IEEE potentials*, vol. 20, no. 1, pp. 11–15, 2001.

[33] S. Pugh, *Total Design: Integrated Methods for Successful Product Engineering*. Engineering technology and design, Addison-Wesley Publishing Company, 1991.

[1] A demonstrative video presentation which visualizes the design method focusing on SECREDAS project [28] can be viewed in the following address: https://youtu.be/j2Ixnkl3EBk.