29th Annual INCOSE international symposium
Orlando, FL, USA
July 20 - 25, 2019

Towards a Model-Based approach to Systems and Cyber Security co-engineering

Juan Navas
Thales Corporate Engineering
juan.navas@thalesgroup.com

Jean-Luc Voirin
Thales Airborne Systems
Thales Technical Directorate
jean-luc.voirin@fr.thalesgroup.com

Stephane Paul
Thales Research & Technologies
stephane.paul@thalesgroup.com

Stephane Bonnet
Thales Corporate Engineering
stephane.bonnet@thalesgroup.com

Abstract. As cybersecurity threats multiply and global public opinion becomes aware of the potential consequences of cybersecurity attacks, customers become more demanding with regard to the proper addressing of cybersecurity concerns in the systems they acquire. As a consequence, systems providers should consider such concerns early in the development life-cycle of their solutions. This paper presents how a model-based approach can contribute to an effective co-engineering effort between cybersecurity and systems engineering during the definition of the system architecture.

Introduction

In the last years, cybersecurity has become a major challenge for worldwide governmental, industrial and service organizations. Attacks have increased in number, diversity and sophistication since the now famous Stuxnet malware targeted at Iran's critical nuclear enrichment infrastructures [Langner 2013]. While organizations are more willing to commit resources in cybersecurity, their efforts seem insufficient: according to the 2018 Thales Data Threat Report [THALES 2018], 36% of the polled companies experienced a successful data breach in the past 12 months, compared to 26% the previous year.

Such a situation tends to deteriorate as we enter into (if not already in) the fourth industrial revolution [World Economic Forum, 2016], in which our dependency on services provided by cyber-physical systems will dramatically increase. As the complexity of these services will rise due to the new and unexpected combinations of systems, the cyber security vulnerabilities and potential targets for cybersecurity attacks will increase as well. Not surprisingly, the INCOSE Vision 2025 [INCOSE 2014] has included security, and particularly cybersecurity, as one of the eight key system characteristics desired by stakeholders. It hence proposes that systems engineers address cybersecurity as a fundamental system attribute that they understand and incorporate into designs.

The way a system shall be protected against cyber threats is determined not only by the context on which it operates, but also by its interactions with external actors, by the properties of the elements composing the system and by how these elements interact. Hence cybersecurity concerns should be addressed from the very beginning of the development process, and considered at each subsequent development stage. Such a "security-by-design" co-engineering approach in which security concerns

are considered at the very beginning of the systems engineering effort, not only diminishes the technical, costs and schedule risks of the project [Honour 2013, Elm 2012], but also permits trade-offs between cybersecurity concerns and other functional and non-functional concerns of the system.

Nevertheless, the implementation of this approach encounters a number of barriers. Indeed, while systems and cybersecurity engineering activities both aim at developing solutions that satisfy stakeholders' expectations, including those related to cybersecurity concerns, co-engineering efforts are hindered by a number of reasons, both internal and external to the enterprise:

- Cybersecurity engineering requires specialized skills and has its own vocabulary, which usually varies following national regulatory frameworks. Acquiring these skills requires a substantial investment, and human resources with both systems and cybersecurity engineering skills are difficult to find.
- Experience shows that cybersecurity engineering activities have their own life-cycle that is more or less uncorrelated with the systems engineering activities. This is mainly due to constraints from certification authorities and need-to-know constraints.

The goal of this paper is to leverage on Model-Based Systems Engineering (MBSE) and propose model-based practices which properly incorporate cybersecurity concerns into the systems engineering activities. We focus on the early stages of this process as they have a strong impact on the subsequent system development activities and on software and hardware sub-systems development activities.

This paper is organized as follows. After a presentation of the background necessary to understand the context, we present a common vocabulary between cyber and systems engineering disciplines, a prerequisite to enable effective communication between both domain specialists; we then present the model-based practices to handle cybersecurity concerns during systems engineering effort; and then we present two techniques to handle the complexity of the co-engineering effort.

Background

Systems & Cybersecurity co-engineering scope

To enable an effective co-engineering effort between cybersecurity and systems engineering, we identified the need of formalizing and tooling-up the interactions between the engineering processes. This is true for all system life-cycle processes [ISO/IEC/IEEE 15288 2015]. In this paper we focus on the process interactions related to those activities leading to the definition of an architectural design of the solution for which we can provide evidence that it properly accounts for cybersecurity concerns. This scope comprises activities from Business or Mission Analysis, Stakeholder Needs and Requirements Definition, System Requirements Definition, Architecture Definition and Design Definition processes of ISO/IEC/IEEE 15288. These activities are led by the Chief Systems Engineer or a similar role according to the organization, supported by engineering specialities experts, including cybersecurity ones.

Figure 1 presents the scope of our proposals. For the sake of simplicity we identify two macro-processes for both cybersecurity and systems engineering streams: (i) the Context and Needs analysis, comprising the engineering tasks leading to a better understanding of the stakeholders expectations and the context in which the system will evolve; and (ii) the Solution design, comprising the tasks leading to the definition of one or more system architectures that are feasible and for which evidence of coverage of stakeholders' needs (including cybersecurity ones) can be provided. This framework is partially based on the Systems Security Engineering Framework presented in [NIST SP 800-160].

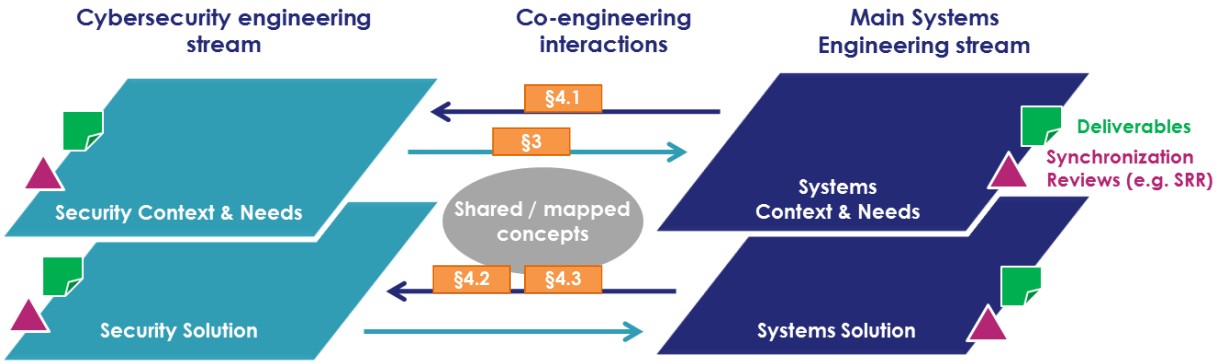

Figure 1: Cyber and systems engineering main interactions

Based on this scope, we identify the main interactions that require formalization and tooling-up:

- As a foundation of the interactions, a vocabulary common to cybersecurity and systems engineering shall be defined to enable effective and efficient collaborative workflows.
- A set of Needs and Context –related interactions aiming at reaching a mature enough definition of the expectations that takes into account the cybersecurity concerns.
- A set of Solution-related interactions aiming at defining the architectural design of the system and providing evidence of the proper consideration of cybersecurity expectations.

The proposals in the two following chapters address the common vocabulary and the model-based practices to handle these interactions.

The use of models on handling these interactions also permits to take advantage of modelling practices to handle the complexity of co-engineering workflows. This complexity may come from the number of participants to the co-engineering effort, the constraints imposed by the context of the project and the regulatory framework and the nature of the system itself, among other factors. These modelling practices are presented in the last chapter.

System modelling with Arcadia and Capella

Arcadia is a model-based method devoted to systems, software and hardware architecture engineering [Voirin 2017]. It describes the detailed reasoning to understand the real customer need, to define and share the product architecture among all engineering stakeholders, to early validate its design and justify it, to ease and master integration, validation and verification. Arcadia can be applied to complex systems, equipment, software or hardware architecture definition, especially those dealing with strong constraints to be reconciled (cost, performance, safety, security, reuse, consumption, weight…). It is intended to be embraced by most stakeholders in system/product/software/hardware definition as their common engineering reference.

Arcadia has been experimented and validated in many real-life contexts for several years. Its large adoption in many different engineering contexts demonstrates an industry-proven comprehensive method for system engineering, capable of adapting to each context in a dedicated manner.

Arcadia intensively relies on functional analysis. It introduces four engineering perspectives (cf. Figure 2): Operational Analysis, System Analysis, Logical Analysis and Physical Analysis. By doing so, it promotes a clear distinction between the expression of the need (covered by the first two perspectives) and the expression of the solution (by the last 2 perspectives).

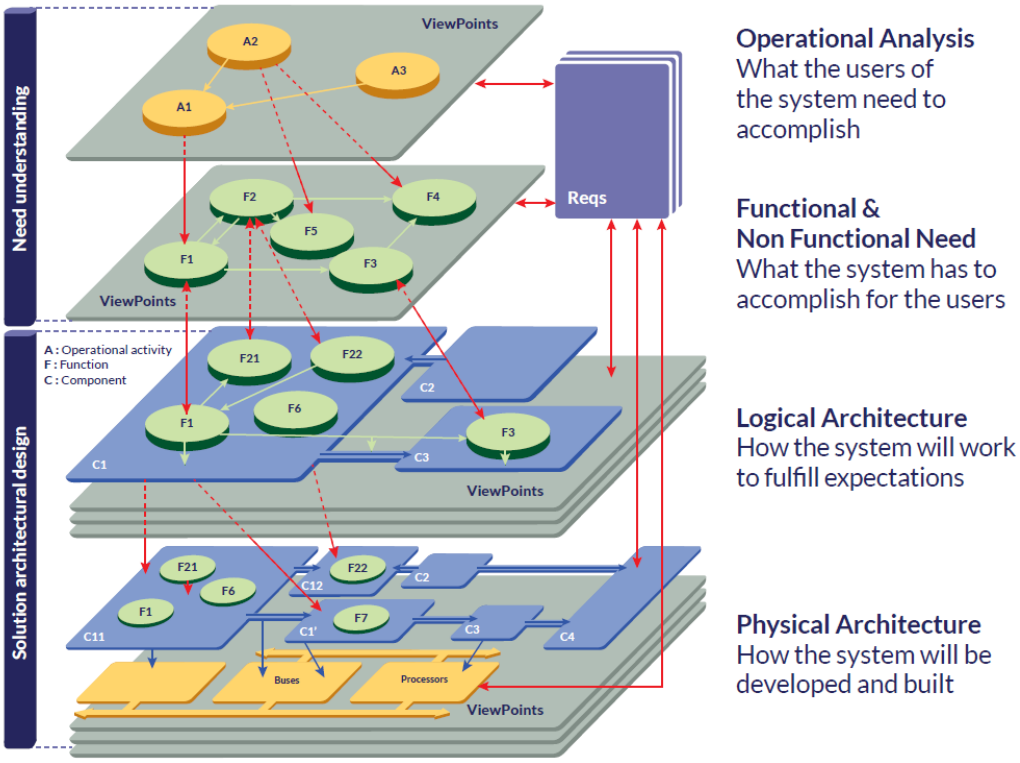

Figure 2: Arcadia engineering phases

The Arcadia method requires a modeling workbench to be effectively implemented. As the lack of properly tailored tools has proven to be a major obstacle to the implementation of MBSE in industrial organizations [Bonnet 2015], Arcadia is recommended to be implemented using the open-source modelling workbench Capella, whose diagrams are inspired from SysML and that has proven suitable for systems engineers with diverse backgrounds and skills [Capella 2017].

Cybersecurity relevant concepts

This chapter briefly presents the cybersecurity concepts that will be exploited in the following chapters.

A *Threat Sour*ce is the intent and method targeted at the intentional exploitation of a vulnerability or a situation and method that may accidentally exploit a vulnerability [NIST SP 800-30].

A *Feared Event* is a generic scenario representing a situation feared by the organisation. It expresses itself by the combination of the threat sources possibly at its origin, a primary asset (see below), a security criterion (i.e. confidentiality, integrity or availability), the related security need and the potential impacts [EBIOS 2010].

An *Asset* is an item, thing or entity that has potential or actual value to an organisation [ISO 55000]. A *Primary Asset* is information or a service deemed important by the organisation; the system security risk manager assesses its security needs. A *Supporting Asset* is an asset supporting primary assets, e.g. information systems, organisations, premises. A *Secondary Asset* is an asset supporting security controls. The system security risk manager assesses the vulnerabilities of supporting and secondary assets [EBIOS 2010].

Security Controls are the management, operational, and technical controls (i.e., safeguards or countermeasures) prescribed for an information system to protect the confidentiality, integrity, and availability of the system and its information [NIST SP 800-30].

Arcadia Systems Engineering relevant concepts

This chapter briefly presents the Arcadia concepts that will be exploited in the following chapters. For a more detailed definition you may refer to [AFNOR 2018] and [Voirin 2017].

An *Operational Entity/Actor* designates a real-world entity involved in operational activities to which the system of interest or its stakeholders should contribute. An actor is a type of operational entity (generally human, usually non-decomposable).

A *Mission* designates a high-level objective of the system. A system *Capability* designates the system's ability to provide a service that supports the achievement of high-level business objectives.

A *Function* is an action, operation or service performed by the system, or by an actor interacting with the system. An *Exchange Item* is a set of elements gathered during an exchange between functions or components (e. g. information, signals, fluids, etc.). An exchange item carries elements with the same transport conditions, simultaneously and with the same non-functional properties. Note that exchanges between functions only express the dependencies between them.

A *Functional Chain* is the specific arrangement of functions and exchanges, forming a path between all possible paths through system data flows, either to describe an expected behavior of the system in a given context, or to express non-functional properties on this path (e.g. latency, criticality, confidentiality, redundancy...).

A *Physical Component* may be a behavioral physical component, designating a constituent part of the system, responsible for implementing some of the functions assigned to the system; or a node, designating a resource hosting behavioral physical components. Finally, a *Physical Link* designates communication or transport means.

Systems & Cybersecurity common vocabulary

This chapter defines a mapping between the cybersecurity-specific concepts and those handled by systems engineering when the Arcadia method is followed.

This mapping is the basis for reaching a common and shared vocabulary between disciplines, as it bounds the co-engineering efforts to a limited scope of discussion. It is also the basis for defining the cybersecurity properties to be attached to the model elements: an model element whose type is mapped to a cybersecurity concept *may* have cybersecurity properties attached to it.

Table 1 provides a summary of the mapping. It only presents a subset of the Arcadia concepts that are mapped to cybersecurity ones. Once a model element has cybersecurity properties issued from the mapping, these can be propagated to other model elements, following the Arcadia meta-model and according to the nature of the system of interest and to design decisions.

Cybersecurity		Systems Engineering (Arcadia)
Threat Source		Operational Entity/Actor
Feared Event		Capability (MisUse Case)
Asset	Primary Asset (service-kind)	Functional Chain, Function
	Primary Asset (information-kind)	Exchange Item
	Supporting Asset	Physical Component, Physical Link
	Secondary Asset	Physical Component, Physical Link
Security Control		Function

Table 1: Mapping between cybersecurity and systems engineering (Arcadia) concepts

Threat sources are mapped to the entities and actors external to the system; they are mapped to concepts of Arcadia's Operational Analysis, as during this step the focus is on the intentions (malicious or not) of these actors. In contrast Feared Events, which represent situations of use of the system that must be avoided, are mapped to Misuse Cases, which are a stereotype of Capabilities in System Analysis (cf. next chapter).

Service-kind Primary Assets are mapped to those Functional Chains representing a service provided by the system, i.e. those involving dependencies between the external actors and the system. As Functional Chains are transversal to the steps of Arcadia method, this choice supports Primary Assets found late in the co-engineering effort or emerging from the analysis of existing system's architectural design, in a bottom-up architectural approach.. Similarly, information-kind Primary Assets are mapped to Exchange Items.

Supporting and Secondary Assets are mapped to Components and Links in Physical Architecture, as they represent tangible assets (components, communication channels) to be protected. Security Controls are mapped to Functions in all Arcadia steps, as protection measures can be applied at all stages of the engineering process.

Impacts of cybersecurity concerns in systems engineering

This chapter presents the model-based practices that are put in place to integrate the cybersecurity concerns into the systems engineering activities. They are presented in a top-down manner, although they may be performed differently according to projects' life-cycles and development strategies.

Note that the figures in this chapter and the following one are only illustrative; they describe in a simplified way either a meteorological balloon (MET) system, or an observation and detection (OBS&DET) system-of-systems which includes MET.

Analysis of the system's cybersecurity context and needs

Systems engineering emphasizes the analysis of the problem before jumping straight to the solution, as a means to develop systems that effectively contribute to the achievement of stakeholders' missions. In Arcadia, this analysis is performed in the Operational Analysis and System Analysis perspectives, and comprises all the elements of the problem space.

Regarding cybersecurity concerns, the model-based analysis of the expectations of the stakeholders will lead to i) the identification of threat sources and other malicious agents, i.e. the entities and actors in the context of the system that may affect system missions, ii) the definition of threat sources goals and intents, and iii) the definition of the mechanisms that threat sources may use to attack the system. This is illustrated in Figure 3.

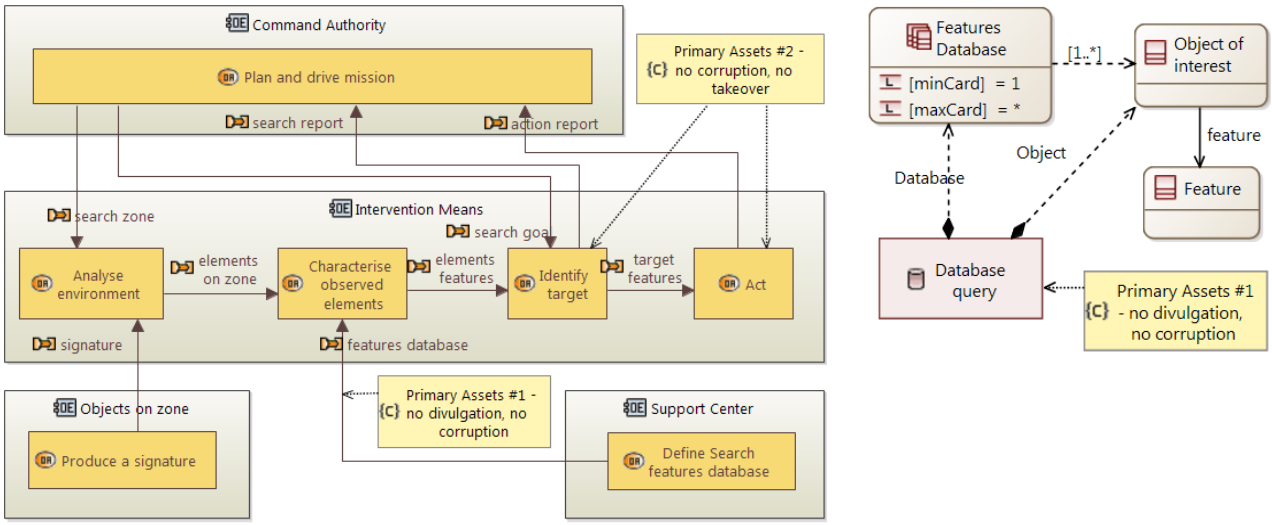

Figure 3: (OBS&DET) Analysis of the expectations of the stakeholders (including threat sources) in system context (left), and of the information been shared (right). Constraints are used to indicate primary assets.

Misuse cases [Sindre 2000, Hope 2004] can be integrated to Arcadia by means of an extension of the Capability concept, as shown in Figure 4. Misuse cases represent the ability of an Actor to provoke a feared situation that may ultimately compromise the accomplishment of system's missions. Misuse cases may follow predefined taxonomies, such as the [STRIDE] threat model. Functional Chains and Scenarios are used to further describe the interactions that may lead to the feared event, as shown in Figure 5.

These model-based practices, when performed in a collaborative way, become the support of the technical dialogue between systems and cyber teams and produce the following results:

- A common and shared comprehension of the operational context in which the system will evolve and of the applicable requirements and constraints
- The identification of the primary assets to be protected: very early as in Figure 3 above (right part: Features Database) and during system requirements definition in a more structured way as in Figure 6 below
- The characterization of cyber security needs and the definition of requirements on confidentiality, integrity and availability that the cyber-protection capabilities of the system shall address
- Multi-criteria evaluations that include cyber security aspects, allowing prioritizing cybersecurity-related requirements in the System Requirements Specification.

These results can feed a formal cybersecurity risks analysis, which is out of the scope of this paper. The risk analysis will establish the set of security controls that will need to be implemented by the cyber-protection capabilities of the system. The model-based practices devoted to the detailed design of these security controls are presented in the next chapter.

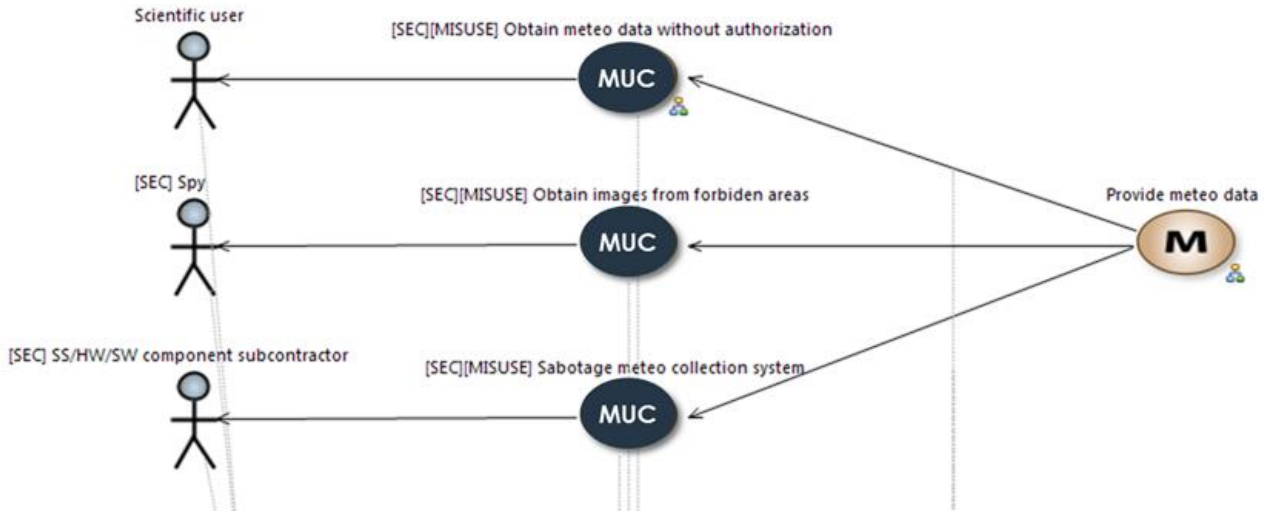

Figure 4: (MET) Three Misuse cases describe how the main mission of the system (provide meteorological data) and its related capabilities can be affected.

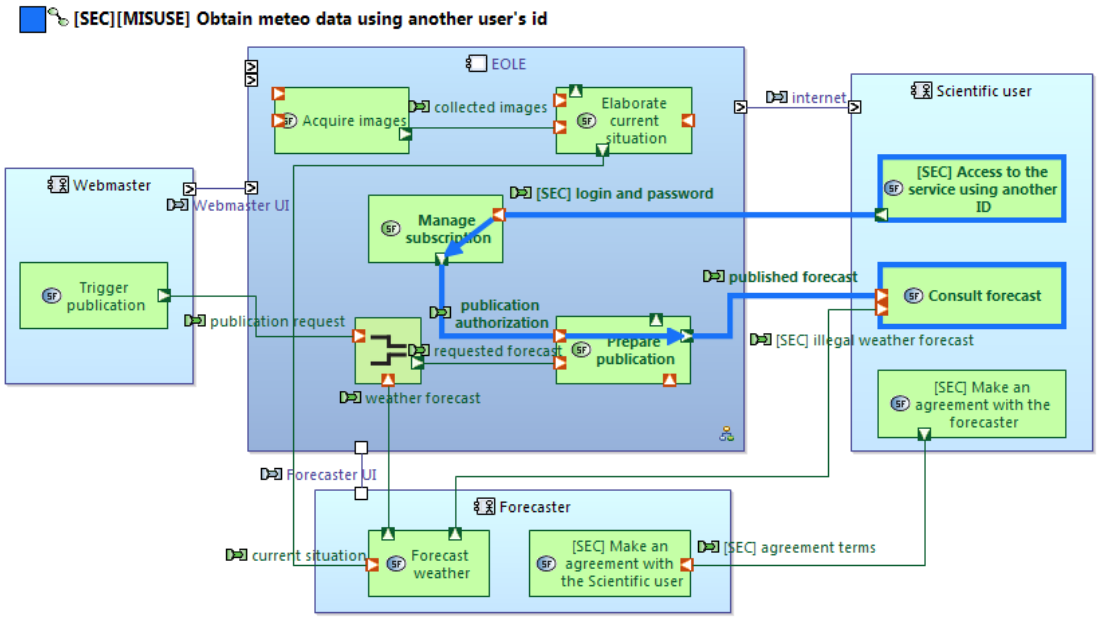

Figure 5: (MET) A Functional Chain (bold blue functional exchanges) describing a Misuse case and the impacted system functions: meteorological data can be obtained by using other user ID.

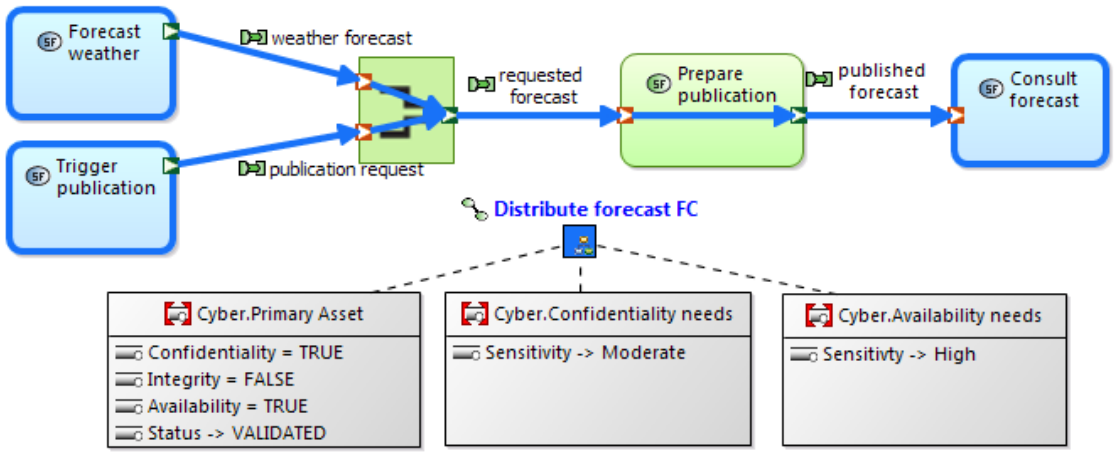

Figure 6: (MET) A service of the system (the “Distribute forecast” Functional Chain) identified as a primary asset and characterized with cybersecurity-related properties

The design of the solution is performed in the Logical Architecture and Physical Architecture perspectives of Arcadia. The former aims at defining a preliminary, technology-agnostic solution that focuses on how the system will behave to fulfill stakeholders' needs. The latter will be the main reference for subsystems and/or components' development teams: it aims at defining the final architecture that takes into account specific technologies and geographical considerations, and at specifying the interfaces between the subsystems and/or components and with the external actors.

A functional analysis is performed on the security controls in order to specify the measures that shall be implemented secure the system. This results on cybersecurity functions implementing monitoring, detection, protection and mitigation measures. These cybersecurity functions have dependencies towards the system architectural components, i.e. they have inputs / outputs that come from / go to other system functions. The integration of cybersecurity functions to the system architecture leads to the definition of "protected" system services that will both (i) contribute to the protection of the system against misuse cases and (ii) realize the system's capabilities and missions. The Figure 7 illustrates this design task.

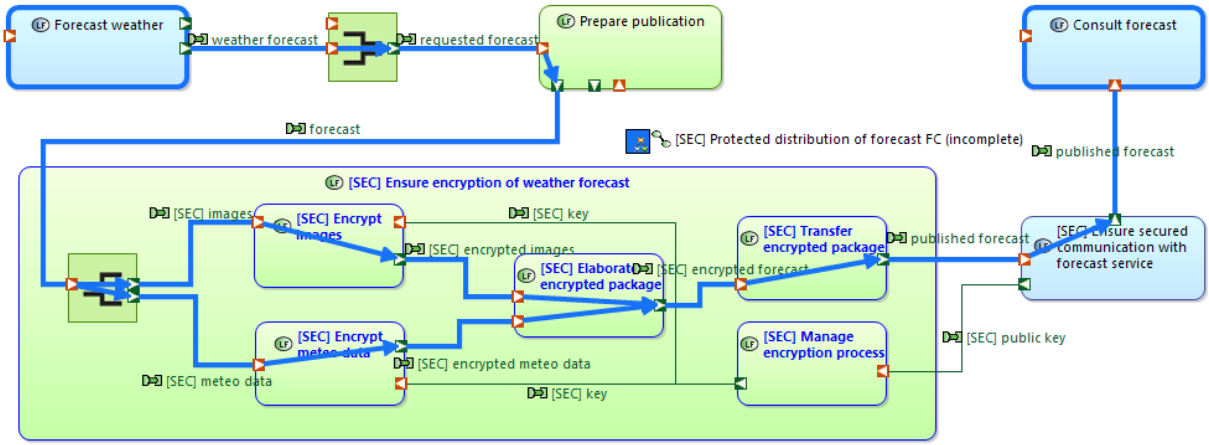

Figure 7: (MET) Development of security controls at Logical Analysis. The encryption security control is refined and integrated with other system functions. The solution-oriented Functional Chain (in blue) represents the protected solution for the forecast distribution service, which is considered as a primary asset.

At this stage, the supporting and secondary assets can be identified. These are the components of the system architecture that implement the primary assets (e.g. contribute to the delivering of a service-kind asset, manipulating an information-kind asset). Their identification is facilitated by the use of models, as the traceability links enable a straightforward retrieval of the functions and components playing a role on implementing the primary assets.

Supporting and secondary assets cybersecurity properties will be set according to the nature of the primary assets and security measures they implement. The automatic coloring of supporting and secondary assets permit to quickly identify them in the overall system architecture, as shown in Figure 8.

These model-based practices lead to the following results:

- A common and shared comprehension of the architecture of the system
- The identification and characterization of the supporting and secondary assets, in an incremental way, beginning with the Logical Components and ending with the fine-grained Physical Components

- Multi-criteria evaluations that include cyber security aspects, allowing taking into account cybersecurity constraints in the process of defining the best possible architecture.

These results can feed a formal cybersecurity risks assessment and risk treatment decisions, which are out of the scope of this paper. These activities will evaluate the adequacy of the architecture and its capacity to secure the system against vulnerabilities exploitation by malicious agents. If required, they may lead to an update of security controls.

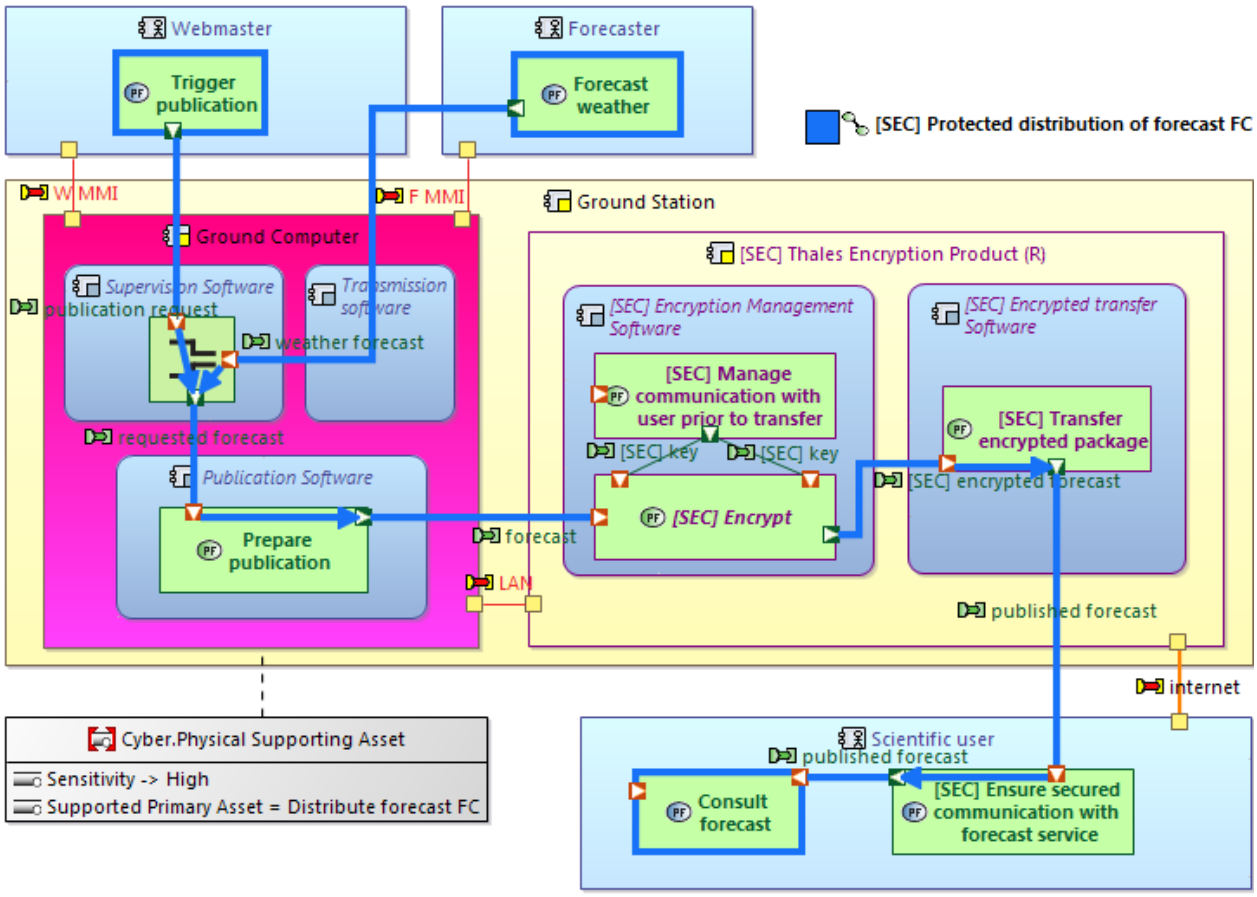

Figure 8: (MET) Physical Architecture view of the cybersecurityaware system. Ground Computer component is a supporting asset, as it implements a service-kind primary asset, and is colored (in pink) according to its cybersecurity properties. Thales Encryption Product and its subcomponents are secondary assets (purple borders) as they implement the security controls presented above.

Early verification and validation of cybersecurity-aware systems architectures

As the architecture of the solution is developed, co-engineering teams continuously check that it remains consistent, complete, valid and accurate. The model-based practices presented above lead to a "by construction" definition of a detailed traceability between the model elements. This traceability can be exploited in multiple ways, including those presented below.

Checking the architectural rules. Model analysis can contribute to checking the way the solution architecture complies with some cybersecurity golden rules and patterns. The definition of the cybersecurity-related rules and patterns to be checked is out of the scope of this paper, but can be illustrated by a few examples:

- If a functional chain is a primary asset, then analyzing the model will automatically identify the components and communication means that must be considered as supporting assets;

furthermore, the consistency of their cybersecurity-related characteristics will be checked as well.

- If each data is characterized regarding its level of confidentiality, then analyzing the model will automatically identify the exchanges that carry this data and all components and communication means involved, on which the consistency of their cybersecurity-related characteristics will be checked.

Checking the coverage of the security controls. The identification of which architecture elements contribute to the satisfaction of cybersecurity-related requirements, and how they contribute, allow co-engineering teams and other stakeholders to gain confidence on the solution architecture. High-level security controls, especially non-functional ones, can be modeled as requirements that can be associated to other requirements and model elements. The semantics of these associations are not fixed and are left to be defined by the architecture team or the organization.

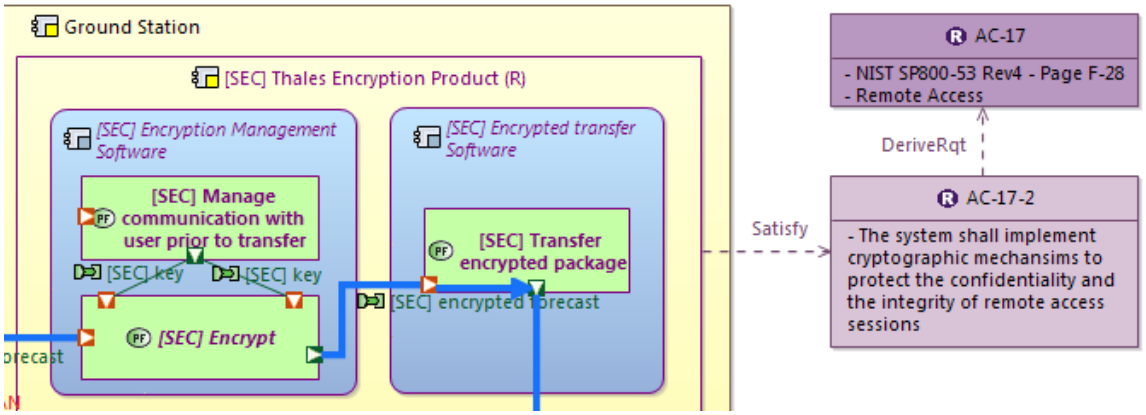

Figure 9: (MET) A physical component defined in the Physical Architecture perspective is said to contribute to the satisfaction of a security requirement defined in System Analysis perspective, which is itself derived from a higher-level security control.

The properties of the architecture elements contributing to the satisfaction of the security controls, as well as the rationale behind these association links, will be inputs to the engineering activities leading to developing the arguments and the body of evidence showing that the system fulfills its security objectives, which is out of the scope of this paper.

Handling the complexity of the co-engineering effort

Cybersecurity is to be engineered as other concerns

The model-based practices presented in the former chapter lead to defining cybersecurity controls that have to be implemented as such. This requires architectural model designers to acquire cybersecurity skills in order to master the integration of cybersecurity concerns into the system (and sub-systems) architecture.

In many cases, this cybersecurity specific engineering effort has to be carried out separately from the main systems engineering effort. This may be due either to the high complexity of the cybersecurity architectural analysis itself, or to confidentiality constraints. In such cases, the cybersecurity architecture will be analyzed and designed using a dedicated model which will be closely related to the model serving for the analysis and design of the system as a whole. The following paragraphs illustrate this approach.

Cybersecurity dedicated models. In most cases, cybersecurity concerns apply to many system assets: e.g, the risk of disclosure may apply to many primary assets, and the number of communication

exchanges that must be encrypted to avoid this can be very high. Similarly, antivirus or boot protection should be applied to any computer in the system, and encrypted VPN to most communications.

Having a dedicated model for cybersecurity concerns allows us to avoid duplication of model elements that lead to unnecessary and very high complexity. For example, in the Figure 10 below, each cybersecurity service (functions here) is related to the primary/essential assets that it is expected to protect in the system model (this is depicted here in the form of constraints for convenience).

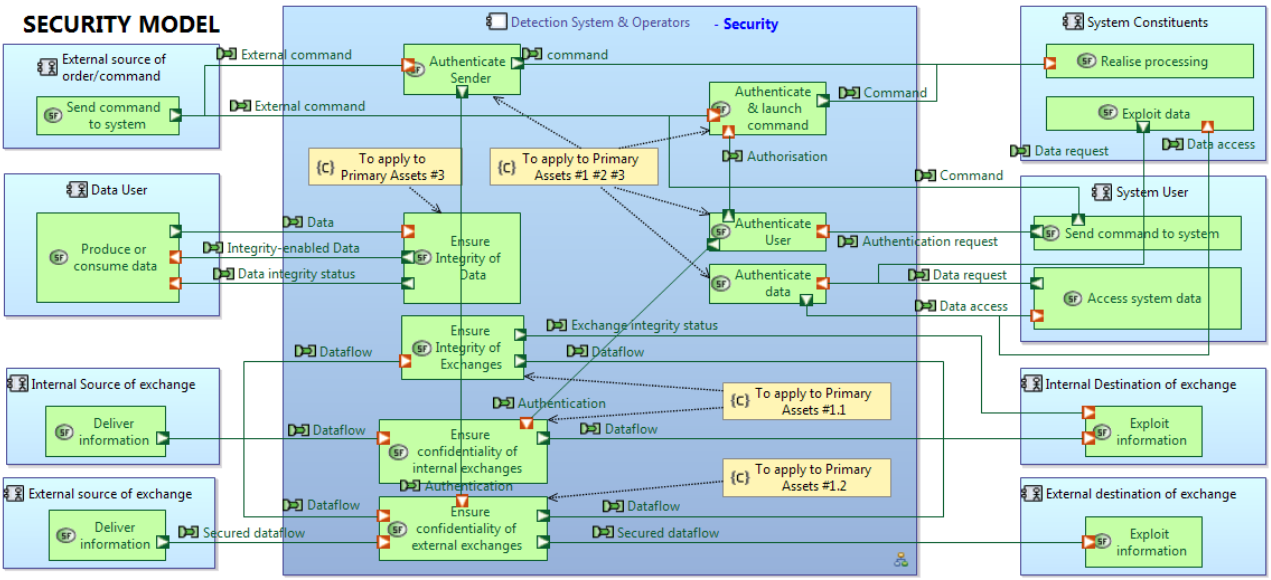

Figure 10: (OBS&DET) A view of a cybersecurity-dedicated needs model – simple example

If this relationship was explicit and *in extenso*, then for a complex system, this would lead to a great number of exchanges between cybersecurity functions and operational elements to be protected, which would be unnecessarily complex, cluttering and costly, with poor added value. Instead, it looks more efficient to:

- In the system model, just characterize each primary asset and the kind of required protection
- In the cybersecurity model, replace all former primary assets by a "representative" generic one (e.g. 'Data User / produce or consume data'), and define expected interactions/exchanges between cybersecurity functions and the generic representative.

The same approach can be applied at the solution perspectives (Logical and Physical Analysis). These describe cybersecurity behavioral components delivering the expected capabilities and services (e.g. software elements, such as encrypted VPN, encryption, monitoring…), and implementation components (e.g. firewall, dedicated boards, and specific cybersecurity assets on each system resource (such as an operational computer). The Figure 11 shows how the "representative" concept has been extended to the supporting assets (Physical Components).

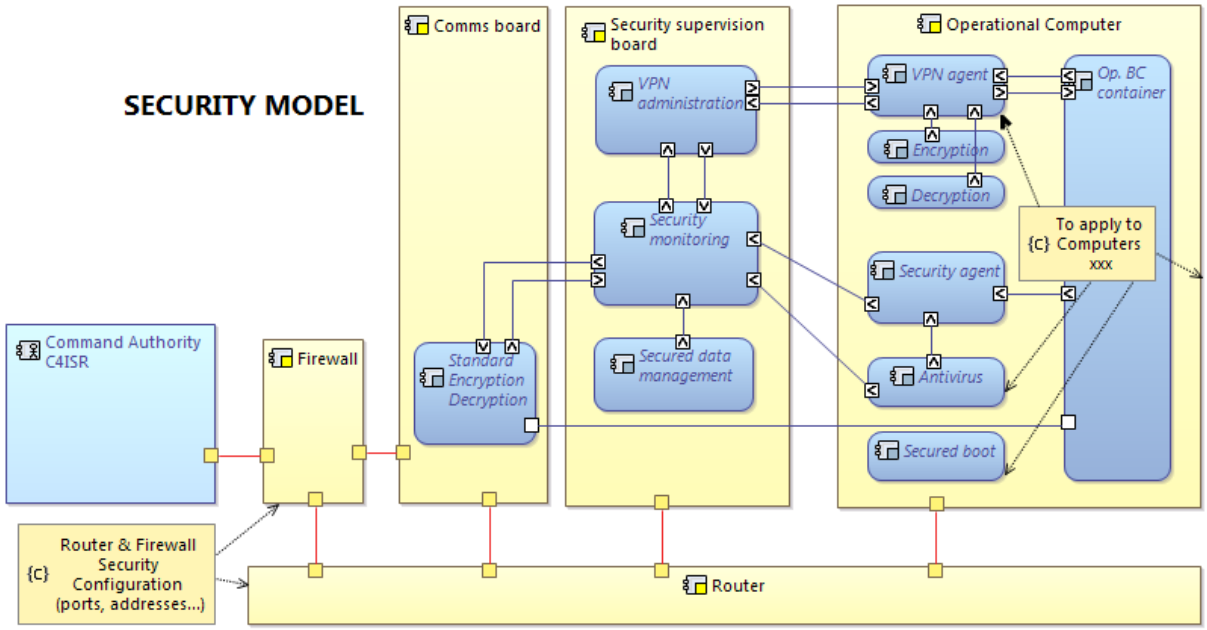

Figure 11: (OBS&DET) The cybersecurity model physical architecture, simplified and partial view. The Operational Computer physical component is a "representative" of several physical components in the system model. A set of behavioral components represent the cybersecurity-specific features that the latter shall implement.

Weaving cybersecurity concerns into the system model. When a cybersecurity-dedicated model is developed, the system physical architecture explicitly mentions the cybersecurity-specific components, but just enough to ensure taking into account constraints from and to these components. Temptation might be to further describe subsystems' cybersecurity architecture details in system model Physical Architecture perspective. This is discouraged as it may lead to level of detail beyond of what systems architects effectively need, and because system engineering and cybersecurity engineering usually have different lifecycles and constraints.

For example, the cybersecurity cabinet is mentioned in the Figure 12 below, because it contributes to interface definitions, wiring constraints, network sizing & performance, etc. Security agents on each computer are mentioned because they impact computing, memory and communications resource consumption. But there is no necessary link between operational behavioral components and Security agents in the system model, if they don't impact architectural decisions at this engineering level.

For the same reasons, the cybersecurity cabinet is not fully detailed, neither are the cybersecurity features on each computer. This would lead to a much more complex model, with increased building and maintenance cost, with little benefit in this case.

One drawback of this approach is the fact that the architectural design of the system is distributed in (at least) 2 models. As a consequence, performing an analysis of how well the system takes into account the cybersecurity concerns becomes a difficult task, especially when only one of the models is available.

This can be mitigated by "resolving" the models at a given point, i.e. integrating the cybersecurity concern into the system model before performing the analysis. However the problem may persist when the approach is generalized to other cross-cutting concerns such as safety, human-factors, etc.

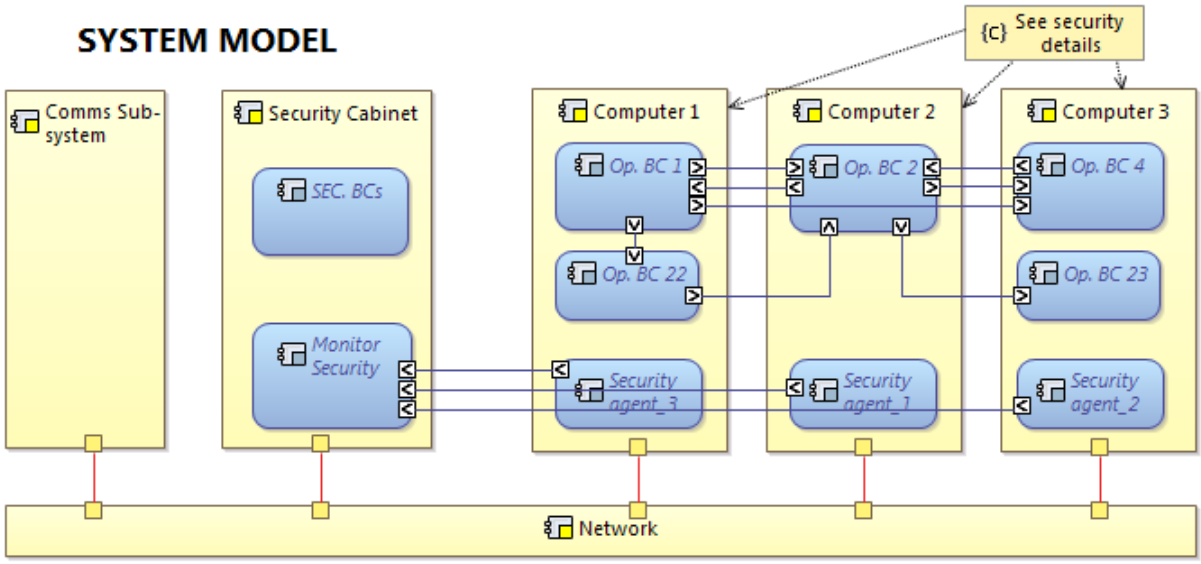

Figure 12: (OBS&DET) The system model implementation architecture, same scope as in Fig. 12

Leverage knowledge using libraries

Knowledge libraries, when developed, organized and managed properly, are powerful tools to improve engineering quality and efficiency, as they permit engineers to reuse and/or adapt well-known and proven architectures to their own system of interest and project's specificities. Regarding cybersecurity and systems co-engineering, libraries can provide benefits:

- During the analysis of needs and context: by leveraging from known threat sources and attacking patterns; by applying standardized security controls, e.g. [NIST 2013]
- During the design of the solution: by adapting existing architectural designs that have proven to be efficient at monitoring, detection, protection and restoration against cybersecurity attacks.

Conclusion

This paper presented a set of model-based engineering practices and techniques enabling an effective co-engineering effort between cybersecurity and systems engineering. These practices are based on a common vocabulary allowing collaboration between engineering domains, and on the Arcadia and Capella systems engineering methodology and tool.

The choice of focusing on a subset of systems engineering processes covering the definition of the architectural design of the solution was motivated by the relevance and the impact of this design on other engineering activities. In the future we plan to address other processes and life-cycle stages of the solution, including the continuous analysis of vulnerabilities during operations and the consideration of operational findings into the architectural design of products and services.

References

AFNOR XP Z 67-140, 2018, 'Information Technology — ARCADIA – Method for systems engineering supported by its conceptual modelling language — General Description – Specification of the engineering definition method and the modelling language'

Bonnet S, Voirin J-L, Normand V , Exertier D, 2015, 'Implementing the MBSE Cultural Change: Organization, Coaching and Lessons Learned', 25th Annual INCOSE International Symposium.

Capella, 2017, Capella Polarsys Website: https://www.polarsys.org/capella/.

EBIOS 2010, 'Expression des Besoins et Identification des Objectifs de Sécurité – Méthode de gestion des risques', Secrétariat général de la défense et de la sécurité nationale française.
Elm J, Goldenson D, 2012, 'The Business Case for Systems Engineering Study: Results of the Systems Engineering Effectiveness Survey, CMU/SEI-2012-SR-009, Software Engineering Institute, Carnegie Mellon University.
Honour E.C, 2013, 'Systems Engineering Return on Investment', PhD Thesis, Defence and Systems Institute School of Electrical and Information Engineering, University of South Australia.
Hope, P 2004, 'Misuse and Abuse Cases: Getting Past the Positive', Building Security In, IEEE Security & Privacy.
INCOSE, 2014, 'A World in Motion – Systems Engineering Vision 2015'.
ISO 55000, 2014, 'Asset management -- Overview, principles and terminology'
ISO/IEC/IEEE 15288, 2015, 'Systems and software engineering – System life cycle processes'.
Langner, R 2013, 'To kill a centrifuge: a technical analysis of what Stuxnet creators tried to achieve', The Langner Group.
NIST SP 800-30, 2012, 'Guide for Conducting Risk Assessments'
NIST SP 800-160, 2016, 'Systems Security Engineering: Considerations for a Multidisciplinary Approach in the Engineering of Trustworthy Secure Systems'
Sindre G, Opdahl AL, 2000, 'Eliciting Security Requirements by Misuse Cases'.
STRIDE, 'The STRIDE Threat Model', Microsoft. https://docs.microsoft.com/en-us/previous-versions/commerce-server/ee823878(v=cs.20)
THALES 2018, 'Thales Data Threat Report: Trends in Encyption and Data Security', Global Edition.
Voirin J-L, 2017, 'Model-based System and Architecture Engineering with the Arcadia Method', ISTE Press, London & Elsevier, Oxford, 2017
World Economic Forum, 2016, 'What is the fourth industrial revolution?', https://www.weforum.org/agenda/2016/01/what-is-the-fourth-industrial-revolution/

Acknowledgements

This research was partially supported by the French CoSS-2 RAPID project.

Biography

Juan Navas is a Systems Architect with +10 years' experience on performing and implementing Systems Engineering practices in industrial organizations. He has worked on the design and the procurement of instrumentation & control systems and simulation systems for petrochemical plants, nuclear fuel cycle plants and nuclear power plants. Today he accompanies systems engineering managers and systems architects implement MBSE and PLE approaches on operational projects, helping them define their engineering strategies, objectives, and guidelines. He holds a PhD on computer science from UBO (Brest, France), a MSc Degree on automation and computer science from MINES ParisTech (Paris, France) and electronics and electrical engineering degrees from Universidad de Los Andes (Bogota, Colombia).

Jean-Luc Voirin is Director, Engineering and Modeling, in Thales Defense Missions Systems business unit and Technical Directorate. He holds a MSc & Engineering Degree from ENST Bretagne, France. His fields of interests include architecture, computing and hardware design, algorithmic and software design on real-time image synthesis systems. He has been an architect of real-time and near real-time computing and mission systems on civil and mission aircraft and fighters. He is the principal author of the Arcadia method and an active contributor to the definition of methods and tools. He is involved in coaching activities across all Thales business units, in particular on flagship and critical projects.

Stephane Paul holds a Master in computer science (1988) and PhD in microelectronics (1991). In 1993, he joined Alcatel to work on object-oriented analysis. Starting from 1996, he was involved in

European R&D projects, dealing with Advanced Surface Movement Guidance and Control Systems (A-SMGCS). From 2000 to 2008, he acted as technical authority on airport systems at Thales ATM's technical directorate. From 2008 to mid-2010, he was head of the Collaborative Technologies Laboratory at Thales Research & Technology. From July 2010, he started research on model-driven security engineering, with a particular focus on security risk assessment, methods and tools. He was also involved in safety & security co-engineering, system of systems engineering (SoSE with UPDM), security for service-oriented architecture, and security solutions for critical embedded systems. He delivers a 3-day cybersecurity engineering course within Thales and intervenes in the Advanced System Architecting and Advanced Engineering courses. He provides risk assessment support to different Thales entities. He also gaves UML courses to Master 2 students and teaches risk management to Master 1 students at ESME Sudria (Paris).

Stephane Bonnet is the Design Authority of the Thales MBSE workbench for systems, hardware and software architectural design. He holds a PhD in software engineering. He dedicates most of his time to MBSE training and coaching activities worldwide, for Thales and other organizations. He helps systems engineering managers and systems architects implement MBSE approaches on operational projects. He is animating networks of experts from all Thales domains and business units to capture operational needs and orient the method and workbench evolutions and roadmaps.