ARTSS: Advanced Resilience Technologies for Secure Service

VPP-COVID-2020/1-0009

WP1: A Method for Development of Resilient and Secure Services

Authors: J. Stirna, I. Stefanišina, J. Grabis, E.Roponena, B. Rajecka, L. Deksne

Version, date: Version 2, 31.03.2021.

Summary

In order to design and deliver secure and resilient services, the ARTSS method has been developed as an extension of the Capability Driven Development. The method provides a structured approach to identification and representation of security and resilience concerns as capability models. The capability models show capabilities and services supporting their delivery as well as capability goals, delivery context and adjustment used to adapt capability delivery in the case of unexpected events such as caused by crisis situations. They serve as a basis for further development of secure and resilient services, digital twins and knowledge sharing. The method is developed as a component-oriented method and seven method components are elaborated.

Contents

1 Introduction

Digital services must be resilient and secure to provide an adequate support in crisis situations. A secure service cannot be used in unauthorized manner and ensures protection and correct processing of users' data. A resilient service is able to quickly restore its normal operations in exceptional circumstances.

A method for development of resilient and secure services (ARTSS method) is elaborated on the basis of the Capability Driven Development (CDD) methodology (Sandkuhl and Stirna 2018). The method provides structured representation of service delivery objectives, performance indicators, context (or operating environment) and security and resilience adjustments. This specification is used to facilitate development and delivery of secure and resilient services. The ARTSS method preserves the main principles and elements of the CDD methodology and supplements these with several new elements and method components:

1. Ecosystem perspective;
2. Digital twin support;
3. Security and resilience goals and knowledge sharing;
4. Learning support.

This paper argues that the digital services provisioning network as a whole as well as individual nodes should possess security and resilience capabilities to ensure safe and sustainable delivery of the services. In this regard capability is seen as *an ability and capacity to meet desired goals (i.e., security and resilience requirements) in dynamic context*. The security and resilience capabilities jointly allow for a quick recovery in the case of security incidents.

The capabilities are developed following the CDD methodology. The methodology defines capability design, delivery and knowledge accumulation processes. Its main features are an ability to capture contextual information in relation to capability objectives and to adapt capability delivery according to changes in the context and performance. It is suitable for design of secure and resilient systems because these concerns are addressed at the organizational and inter-organizational level rather than just the technical level. It a component-oriented methodology and extensions can be developed for specific purposes.

This deliverable elaborates a method extension of the CDD methodology for development of security and resilience capabilities. The method extension is specifically targeted towards networked organizations. Seven method components are described and an example of capability modeling for the secure and resilient computer networks is provided. The capability models developed using the ARTSS method focus on resilience and security aspects. The capability delivery also requires a set of other services and these can be create using other suitable service engineering methods. The ARTSS method extension of the CDD methodology is first described in the paper Grabis et al. (2020).

The rest pf the deliverable is structured as follows. Section 2 describes the method development approach. The overview of the ARTSS method is given in Section 3. The method components are elaborated in Section 4. Section 5 introduces tools supporting the ARTSS method. A modeling example is provided in Section 6 and Section 7 concludes.

2 Method Development Approach

Capability Driven Development methodology consists of a number of method components each focusing on a specific task of the capability cycle, such as Capability Design, Context Modeling, Patterns and Variability Modeling, and Capability Adjustment Algorithm

Specification. These method components are considered to comprise the regular CDD method, extended with method extensions for dealing with certain business challenges such as supporting business process outsourcing, context-aware configuration of e-government services, industrial symbiosis, designing of new entrepreneurial ventures (Sandkuhl and Stirna, 2018), as well as managing service configuration with the support of open data (Kampars et al, 2020). Method components and method extensions are described following the structure of a method component initially outlined in (Goldkuhl et al. 1998).

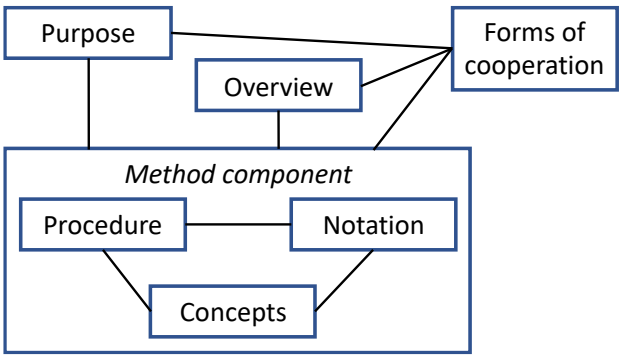


Fig. 1. Overview of the structure of a method component, adapted from (Goldkuhl et al. 1998).

Each method component or extension is to be described according to the structure shown in Fig. 2. More specifically:

1. *concepts* specify what aspects of reality are regarded as relevant in the modelling process, what is important and what should be captured a model.
2. *procedure* provides guidance how to identify the concepts in practice, prerequisites and resources. It should be described in terms of steps to be performed with input, output and tool support.
3. *notation* specifies how the result of the procedure should be documented, i.e. graphical representation of concepts including relationships.
4. *overview* describes the relationships between the individual method components, i.e. which components are to be used and under what conditions, as well as the sequence of the method components (if any).
5. *forms of cooperation* describe the necessary skills, ways of cooperation, and organizational roles that should be involved in using the method component.
6. *purpose* states what the purpose of the method is, what modelling or problem-solving task can be addressed by the method component.

The benefits of structuring a method in method components are: (i) the resulting method is not a monolith which means that its parts can be combined and applied depending on the needs of the modeling domain; (ii) all components are based on a common meta-model which allows efficient elaboration of new components and extensions, (iii) method components or their parts can be easily updated and replaced, e.g. the goal modeling language of CDD is based on 4EM, and can be replaced with another; (iv) the method can be extended with other methods offering modelling perspectives that are not currently addressed by it.

3 Overview of the ARTSS Method

The ARTSS method provides a systematic and structured way to design resilient and secure services. It is based on the ARTSS approach of combining capability driven development, digital twin, knowledge management and learning support in order to develop and deliver

resilient and secure services. The method can be tailored to the needs of specific organizations and it is compatible with service engineering processes at these organizations.

3.1 ARTSS approach

Built-in resilience and citizen protection are a must for delivering essential services in a crisis situation such as the Covid-19 pandemic. The ability to continuously monitor and to adapt the delivery of these services contributes to their resilience. To this end the ARTSS project aims to develop a method that is based on the following pillars (Figure 1):

- Business ecosystem modeling to map the diverse actors and their contributions involved in the service delivery
- Capability management to design and run context dependent adjustment and management of services.
- Digital twins to ensure that the adjustments (decisions and actions) are propagated to the service delivery including advanced visualization of the service ecosystem.
- Large volumes of contextual data (live as well as historical) processing and management
- Accumulation of reusable crisis response knowledge (best practices) in a form of pattern.
- Secure and resilient service adoption approach supported by digital learning material to facilitate broad-scale adoption of the ARTSS results.

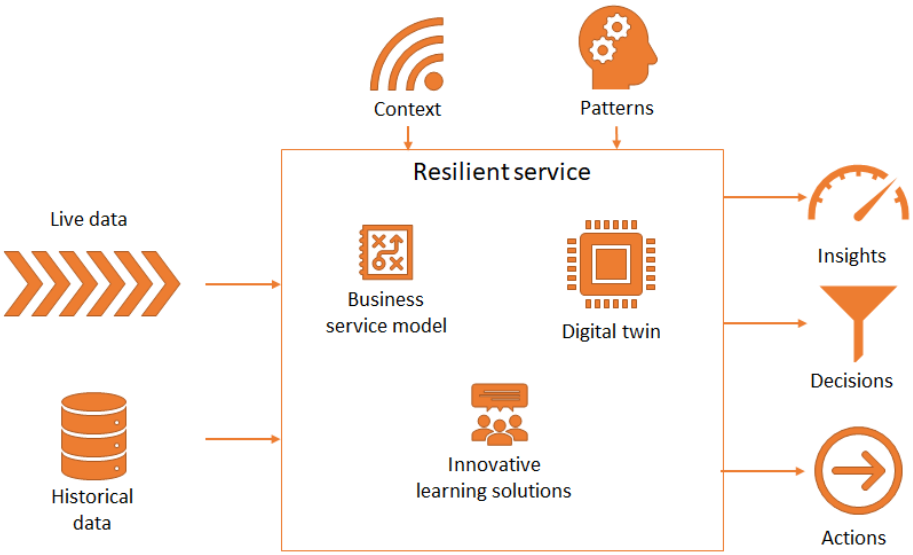


Figure 1 ARTSS approach.

3.2 Concept Model

Capability defines an ability and capacity of the network and its nodes to provide secure and resilient services. The network has common goals, exchange contextual information as well as shares knowledge to a degree of trust required in the network. Figure 2illustrates additional concepts introduced in the CDD meta-model to design secure and resilient service delivery capabilities. The network is required to possess security and resilience capabilities. Delivery of these capabilities is supported by services (used as an alternative implementation

mechanism to processes in the original CDD meta-model). These services are provided by a network consisting of multiple nodes. The Security concern concept is introduced to identify goals dealing with security and resilience. In order to explore network behavior depending on contextual changes and adjustments used, its virtual representation or digital twin is used. It uses live data to diagnose potential problems and to plan future capability delivery. It plans application of adjustments to optimize capability delivery according to the specified goals. The digital twin uses actual network services in its simulations. It is not aimed to represent the whole network and its services in the digital twin. The digital twin focuses on the security concerns identified during the capability design.

Patterns define knowledge about development and delivery and resilient services. They are extracted during capability design and delivery and are used to identify a course of action in the case of specific context situations. A learning module is a specific type of pattern and supports learning about proper usage of services to attain security and resilience benefits.

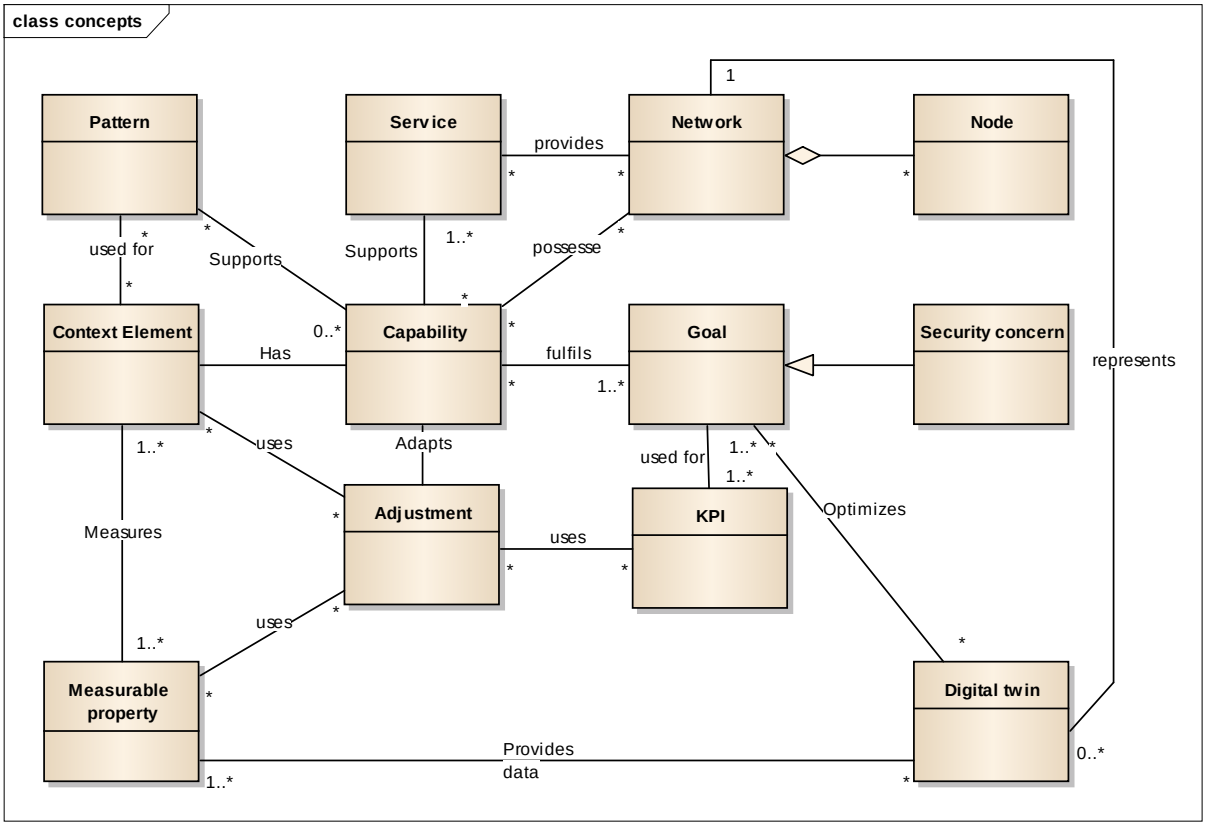


Figure 2 The concepts model of the ARTSS method.

3.3 Key Concepts

The key concepts of the ARTSS method are described in Table 1. The method components use these common concepts and additional concepts are introduced if necessary.

Table 1 . Key concepts of the ARTSS method

Concept	Description
Adjustment	An algorithmic recommendation to adapt capability delivery according to the context situation
Capability	Ability and capacity to achieve organization's objectives in variable contextual situations
Context Element	Represents information characterizing situation of an entity, i.e., service
Digital twin	Virtual representation of real life objects for monitoring and control purposes
Ecosystem	Parties involved in consumption and provisioning of secure and resilient services
Goal	A desired state of affairs that needs to be attained
KPI	Measures achievement of the goals
Learning module	Supports learning about resilient and secure services
Measurable property	Measurements of the contextual situation (i.e., data used to evaluate the context)
Network	A network of parties involved in providing resilient and secure services
Pattern	Structured reusable knowledge about development and provisioning of resilient and secure services.
Security concerns	A specialization of the goal concept to represent resilience and security concerns
Service	A software component providing a specified component in response to the consumer request

3.4 Participants

The participants involved in the development and provisioning of secure and resilient services are (Figure 3):

- Consultant – an expert of the ARTSS method who design capability driven resilient and secure services;
- Capability owner – enterprise or organization possessing or striving to possess a capability of providing secure and resilient services;
- Service consumer – service users;

- Data provider – provider of data necessary for evaluation capability delivery context. That includes both internal and external data providers. The usage of open data sources is emphasized;
- Service provider – a provider of secure and resilient services.
- Infrastructure and tools provider – a provider of technical means for developing and provisioning of secure and resilient services.
- Sage – manages the pattern repository.

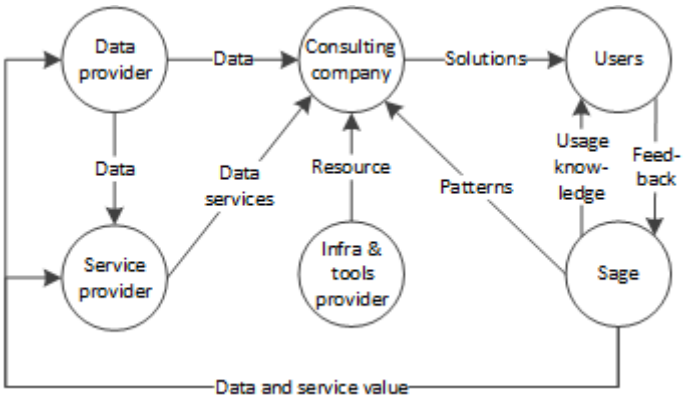


Figure 3 Participants of the secure and resilient service delivery ecosystem.

3.5 Method Components

Seven method components are defined as a part of the ARTSS project:

1. Capability scoping – to identify and review capabilities, which are necessary to deliver secure and resilient services in changing contextual situations;
2. Service definition – to define services required to deliver capabilities with emphasis on services dealing with security and resilience aspects;
3. Capability definition – to elaborate a detailed capability model;
4. Digital twin design – in order to continuously monitor and control capability delivery, a digital twin is created. The method component describes digital twin design according to the capability model;
5. Pattern identification – new patterns are identified by analyzing capability models developed in different uses cases;
6. Pattern consumption – search for the patterns suggesting solutions for dealing with resilience and security concerns during capability delivery and service provisioning;
7. Capability ecosystem – gradual evolution of capabilities in an ecosystem of service providers and consumers.

The list of components can be extended to include additional method components or specific service development and delivery need.

Figure 4 gives an overview of the main elements of the ARTSS method including method components, key concepts and tools. The ARTSS method does not specify a particular sequence of deployment of the method components.

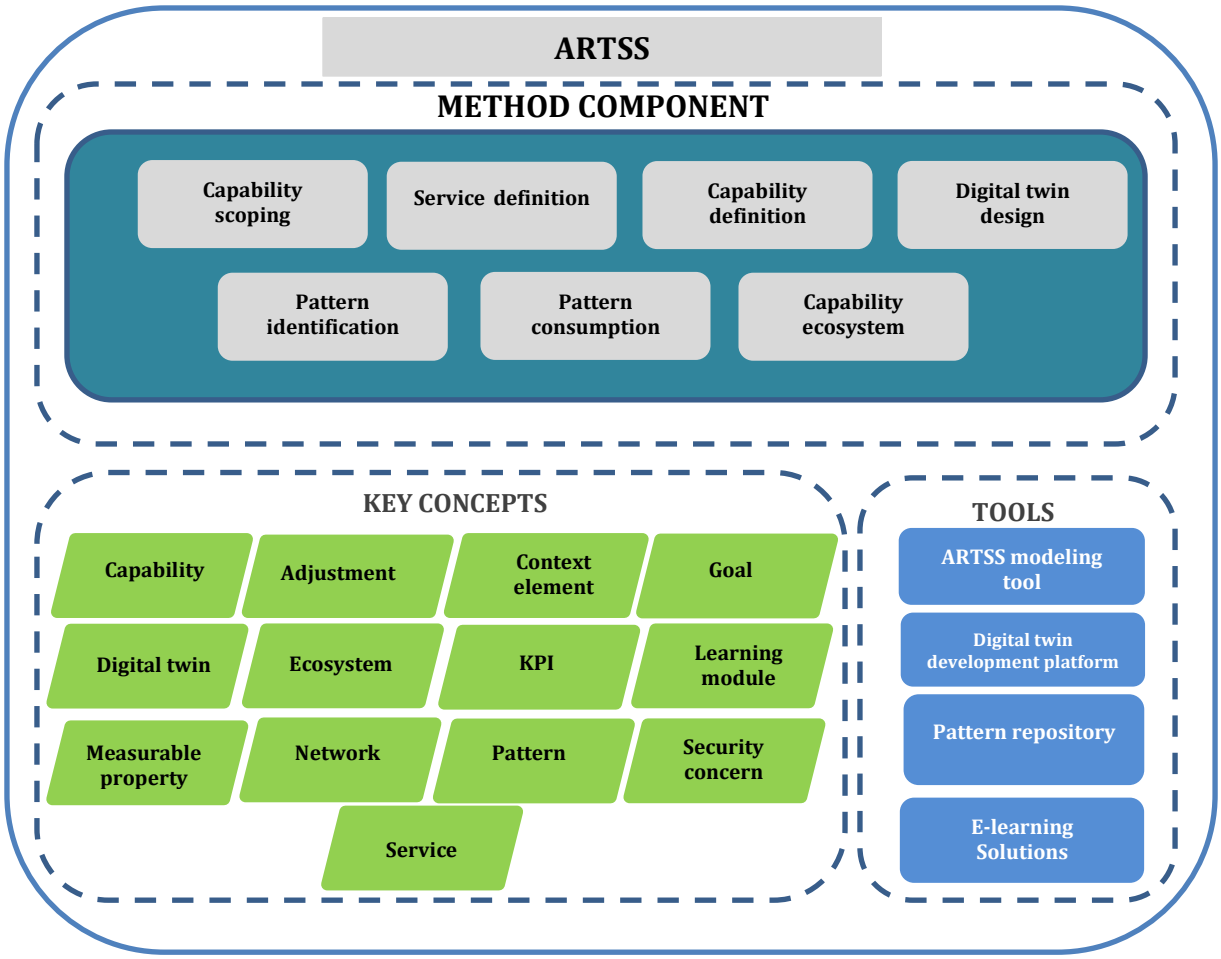


Figure 4 . Key elements of the ARTSS method.

3.6 Related work

Design of networks providing digital services is a challenging research area (De Reuver et al., 2018). Security concerns are particularly prevalent in such networks often due to their obscure nature (Mouratidis et al., 2016). Specialized modeling methods help to identify and highlight the security concerns (Elahi and Yu, 2009). Security Incident Response Modelling Language is proposed to model security and recovery issues (Athinaiou et al, 2018). Various aspects of designing secure systems can be captured using a unified modeling technique representing attack trees, vulnerability cause graphs, security activity graphs, and security goal indicator trees (Byers and Shahmehri, 2010). Lack of transparency in networks also can be addressed by means of data analysis (Lu et al., 2013).

Similarly, resilience also should be a measurable network design feature (Fiksel 2003). Considering design of resilient systems, the study (Bodeau and Graubart 2017) suggested starting from a goal-framework for achieving: diversity (variety of actors), efficiency (productivity and resource utilization), adaptability (transparency and flexibility), and cohesion (alignment of actors and their capabilities), which need then to be realized by the goals and objectives specific to a business domain. ISO 22316:2017 provides the guideline for any size or type of organization. It is not specific to any industry or sector and it can be applied throughout the life of an organization. MITRE (Korpela et al., 2013) has outlined a set of “design principles” for cyber resilience elicited from various domains including

Evolvability, Survivability and Security. In (Ross et al. 2019), the National Institute of Standards and Technology (NIST) provides a framework for improving the cybersecurity and resilience of critical infrastructures. Some research studies have proposed the guidelines organizational cyber resilience and measuring its maturity by extending the design of the information systems with, for example, additional interfaces to the systems' environment for increased situational awareness, design of alternative, modular software capabilities, development of security-related parameters and related metrics, and rich testing to different conditions (Fiksel, 2003). Haque et al. (2019) present a comprehensive cyber resilience framework for Industrial Control Systems by decomposing "resilience" into a hierarchy of several sub-metrics. Their resilience framework can serve as a platform for a multi-criteria decision aid and help technical experts in identifying the gap in the study of network resilience.

The concept of digital twins is starting to get attention for application in the security management domain. For instance, Eckhart et al. (2019) propose to use digital twins to rise cyber situational awareness for cyber-physical systems through visualizations. Viability of using data streams in digital twining has been recently demonstrated in (Kritzinger et al., 2018). In advanced cases (Murphy et al., 2020) digital models of service networks are created automatically (by analyzing large sets of data available in the existing enterprise information systems).

4 ARTSS method components

The method components are elaborated for specific need of development and provisioning of secure and resilient services. They are described following the method development approach presented in Section 2.

4.1 Method Component: Capability Scoping

Method Component Name: Capability Scoping

Usage objective: Review and identification of capabilities necessary to achieve organization's objectives in volatile operating environment. During the capability scoping process, a group of experts work in a collaborative environment to create an initial capability model. The initial model is created using a sub-set of capability design concepts following an informal modeling approach (Figure 5)

Concepts: The method component uses a subset of the standard capability modeling concepts:

1. Capability
2. Context element
3. Measurable property
4. Goal
5. KPI
6. Adjustment

Notation: The key concepts are represented informally using concept specific stickers.

Figure 5 . Capability scoping notation

Method process:

Input data:

1. Use case description
2. Enterprise architecture and enterprise models if available

Activities:

1. Capability naming – the capability name is identified. The name emphasizes the security and resilience aspects if possible.
2. Identification of goals and context elements – brainstorming on relevant goals and context elements by adding stickers to the workspace.
3. Identification of measurable properties and KPI for the context elements and goals, respectively. KPI emphasize the need to clearly measure the goals. The measurable properties emphasize feasibility of evaluation of the context with respect to data availability.
4. Identification of adjustments – indicate mechanisms for adapting the capability delivery in the case of security and resilience concerns.
5. Revision of the initial model.

Results:

Capability map including the base definition of the capability and its key elements such as context, goals, KPI and adjustments.

Tools: Collaboration tool – online collaboration tool not requiring special training, e.g., Miro.

Participants:

1. Capability owner
2. Service consumer
3. Consultant

4.2 Method Component: Service Definition

Method component name: Service definition.

Application objective: Services provide software functionality to deliver capabilities. A set of services create capability delivery application. The method component describes identification of services needed for capability delivery. The main attention is devoted to services strongly influencing the overall resilience and security of the system. The capability delivery also requires a number of other services to execute business processes without specific emphasis on resilience and security. These services are developed using traditional service engineering techniques (Figure 6).

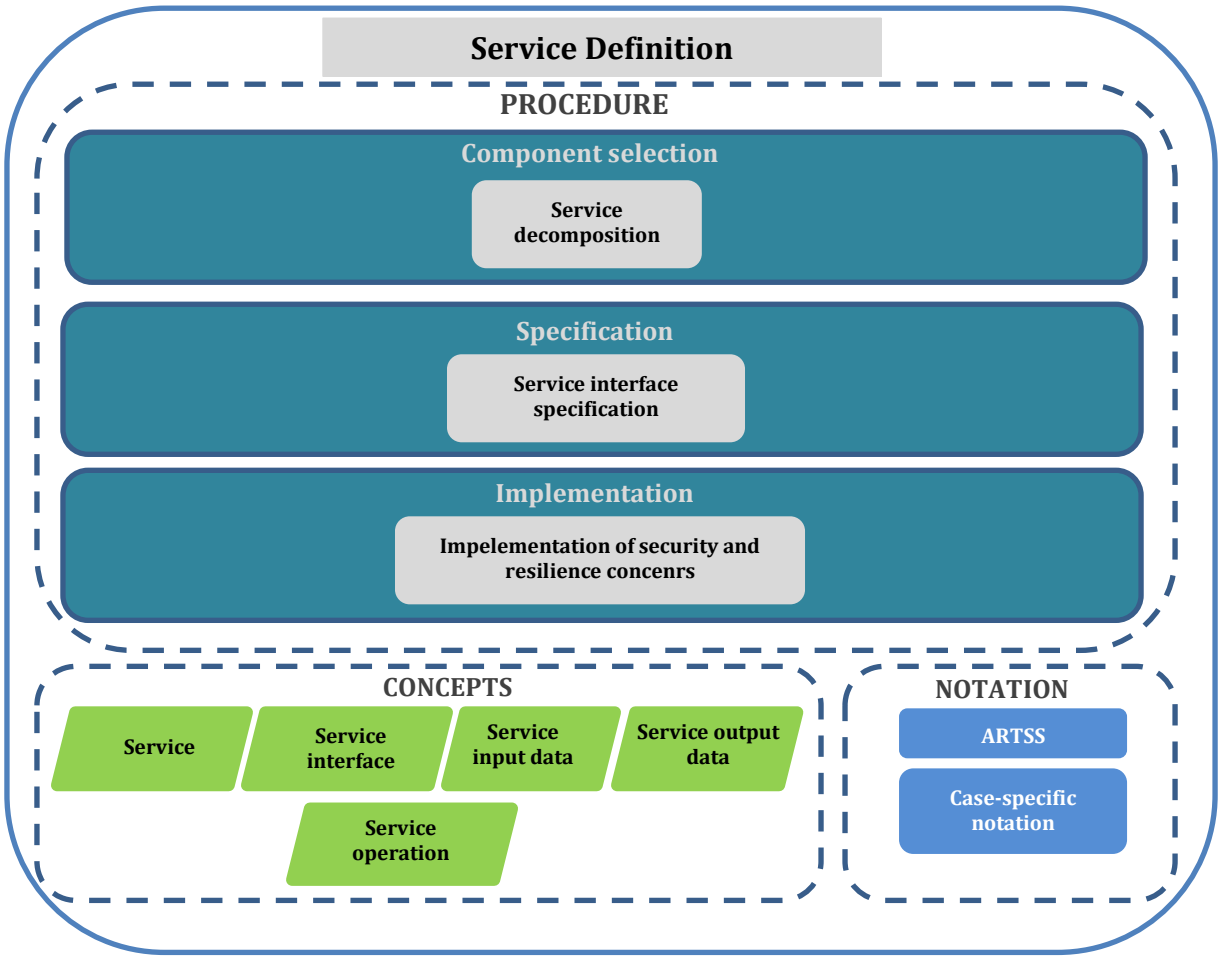


Figure 6 . Resilient and secure service definition method component.

Main concepts:

1. Service – a software component providing specific functionality and available as a service in response to the customer request.
2. Service interface – machine-readable specification describing the service and enabling its invocation
3. Service input data – service arguments, which are passed to the service upon its invocation. The capability model is one of the input parameters. That allows the service to gain access to the elements of the capability model including adjustments, KPI and context elements.
4. Service output data – the service response.
5. Service operation – functions provided by the service.

Notation:

1. ARTSS notation.
2. Case-specific notation - according to the company specific engineering process

Process:

Input data:

1. Enterprise architecture.
2. Documentation of services – documentation of existing services.
3. Machine-readable capability model.

Activities:

1. Service decomposition – capability is delivered using a set of ICT solutions. Individual components are distinguished as services if they are related to the elements of the capability model such as KPI, context elements or adjustment or deemed important to ensure capability delivery resilience and security.
2. Service interface specification – in order to expose service to consumers, its interface is defined including definition of input and output data and operations provided. The service uses the capability model as one of the inputs. Therefore, the capability model itself is published as a service.
3. Implementation of security and resilience concerns – resilience and security features are added during implementation of the service. That is made possible by optimizing the service performance according to the capability goals and using context data and adjustments to adapt service to address resilience and security concerns.

Results:

1. Representation of services in the ARTSS model.
2. Published service interface documentation.
3. Service implementation.

Tool support:

1. ARTSS modeling tool.
2. Service specification tool – e.g.., OpenAPI in the case of REST services.
3. Service implementation tool – services are implemented using tools according to the company's software engineering processes;
4. Service deployment environment – services are deployed in a containerized environment, e.g., Docker.

Stakeholders:

1. Service provider;
2. Consultant;
3. Infrastructure and tool provider.

4.3 Method Component: Capability Definition

Usage objective: Develop a formal model for the identified capabilities. The level of details should be sufficient for capability management, service implementation and digital twin

design purposes (Figure 7). This method component is directly derived from the previous work on Capability Driven Development (Sandkuhl and Stirna, 2018)

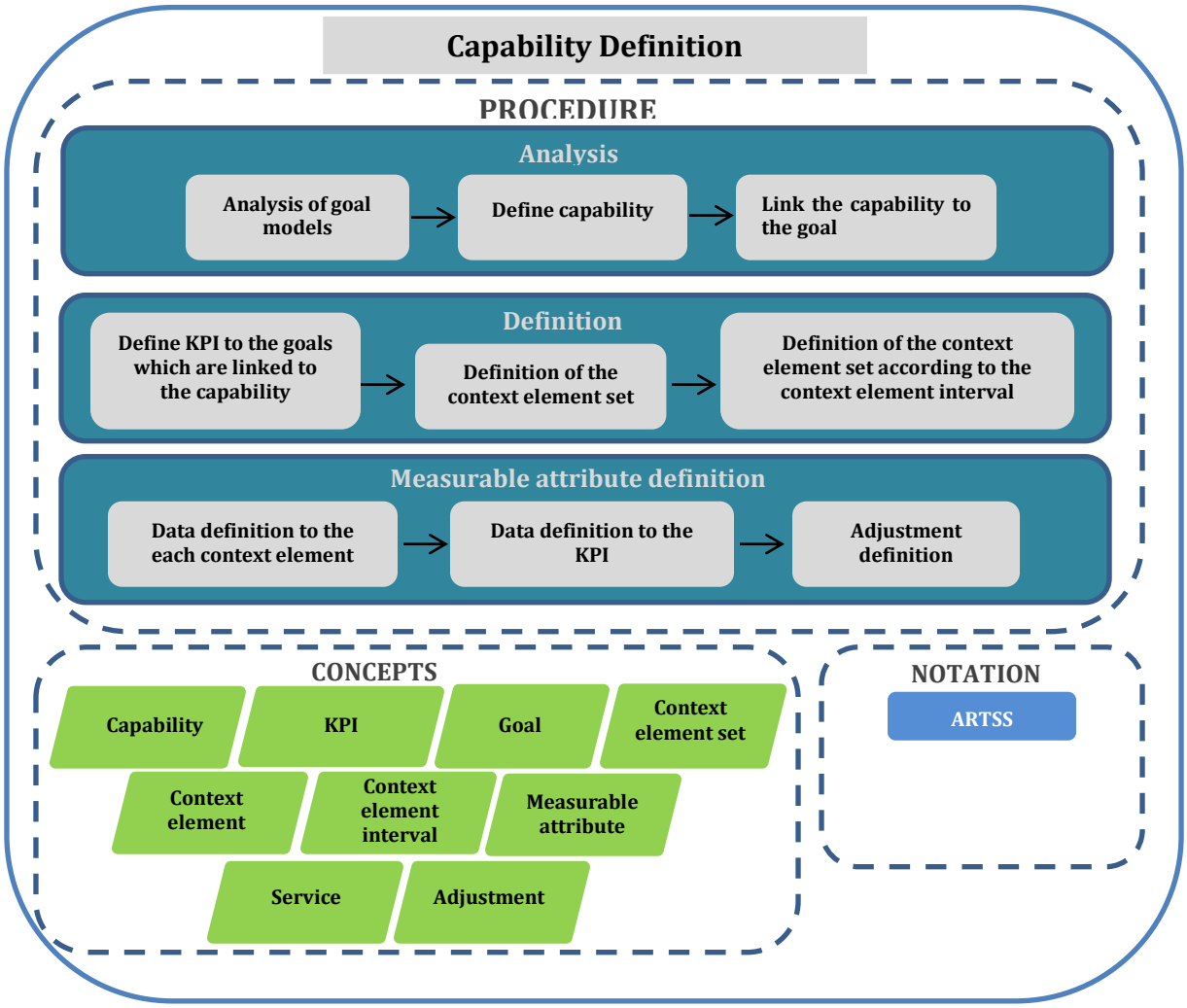


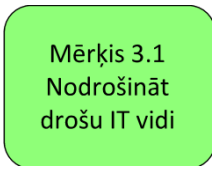
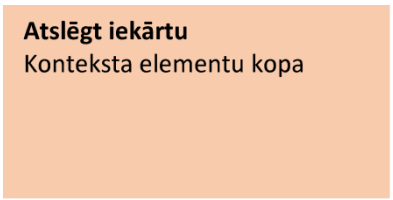
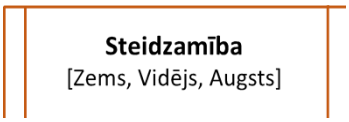
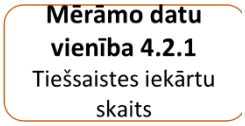
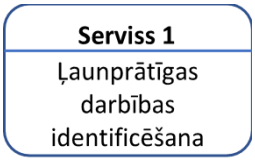
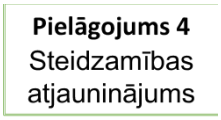
Figure 7 . Capability definition method component.

Main concepts:

Goal, KPI, capability, context element, context element set, measurable attribute.

Notation: ARTSS notation

Capability	**Spēja 1.1** Atslēgt iekārtu
KPI	KPI 3.3.1 Brīdinājumu skaits mēnesī

Goal	
Context element set	
Context element	
Context element range	
Measurable property	
Service	
Adjustment	

Procedure:

Input data:

1. Enterprise models – existing enterprise models if any. The models describe vision, strategy, business processes and data.
2. Capability map
3. Business services
4. Data models – describe data entities used in enterprise information systems.

Goal:

To elaborate a detailed model for every capability

Activities:

There are several ways to perform capability modeling. The procedure could be started by goal analysis, business process analysis or conceptual analysis. These pathways are described in (Henkel et al., 2018). Regardless of the pathway chosen the following tasks are performed:

1. Existing enterprise goal models are analyzed to identify capabilities required to achieve the goals;
2. Define capabilities corresponding to the goals and associating the goals with the capabilities;
3. Define KPI for measuring the goals
4. Define context elements defining the capability delivery context
5. Define context element ranges within which the capability is deemed suitable
6. Define measurable properties for each context element;
7. Define capability adjustments and associate these with context elements affecting the adjustments;
8. Mark resilience and security goals;
9. Identify adjustments, KPI and context elements critical to resilience and security by means of analysing their connections to the resilience and security goals.

4.4 Method Component: Digital Twin Design

Methods component name: Digital Twin Design

Usage objective: In order to ensure continuous monitoring of resilience and security as well as to control the service and capability delivery, a digital twin is created for the service delivery network. The method component defines design of a digital twin according to the capability model.

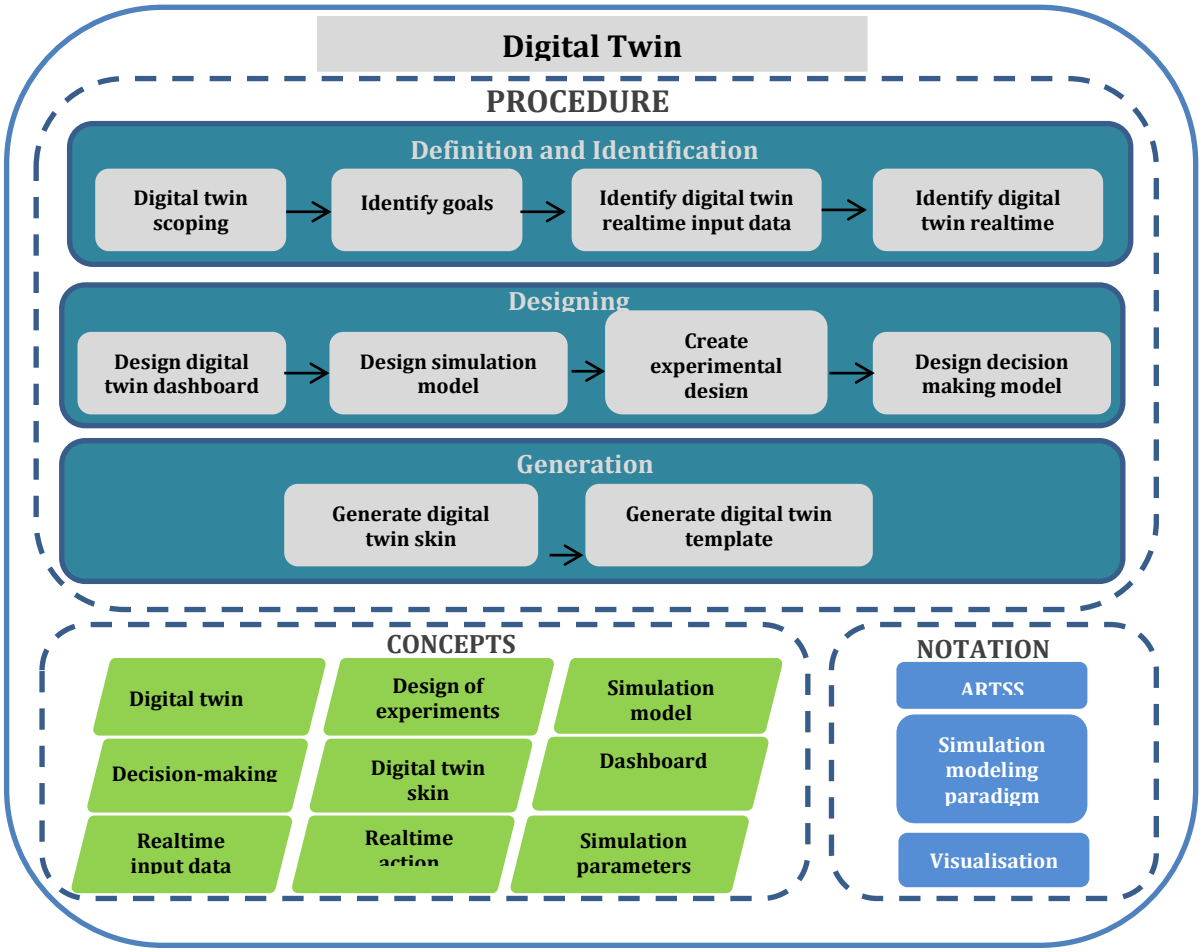


Figure 8 . Digital twin design method component.

Key concepts:

1. Digital twin – a virtual representation of real world object;
2. Design of experiments – resilience and security evaluation scenarios for experimental evaluation according to the capability's goals
3. Simulation model – used for purposes of dynamic analysis of network of services enabling capability delivery
4. Decision-making – selection of the real-time actions on real world objects according to the simulation results.
5. Digital twin skin – digital twin visualization front-end
6. Dashboard – the dashboard contains the KPI defined in the capability model and relevant in the context of the digital twin
7. Realtime input data – input data are measurable properties and context elements defined in the capability model and relevant in the context of digital twin.
8. Realtime action – real-time actions on real world objects invoked by the digital twin and corresponding to adjustments in the capability model;
9. Simulation parameters – simulation parameters specified in the experimental design and need for experimental evaluation of resilience and security of the services.

Notation:

1. ARTSS notation
2. Simulation modeling notation according to the simulation modeling paradigm used
3. Visualization notation

Process:

Input data:

1. ARTSS model
2. Historical data for simulation modeling purposes

Activities:

1. Digital twin scoping – select services requiring design of a digital twin from the capability model.
2. Identify goals relevant to the digital twin – select digital twin relevant goals in the capability model.
3. Identify digital twin real-time input data – select measurable properties and context elements in the capability model needed in the digital twin.
4. Identify digital twin real-time actions – identify adjustments in the capability model used as actions in the digital twin.
5. Design digital twin dashboard – identify KPI from the capability model to be represented in the digital twin dashboard
6. Design simulation model – a simulation model is developed according to a suitable method. The simulation model will be used by the digital twin for experimentation and decision-making purposes.
7. Create experimental design – to specify simulation modeling scenarios for detailed exploration.
8. Design decision making model – to design a decision-making algorithm for selecting a suitable adjustment or real-time action according to the simulation results.
9. Generate digital twin skin – generate visualization of digital twin.
10. Generate digital twin template in a target development environment

Results:

1. Digital twin skin
2. Digital twin template

Tool support:

1. ARTSS modeling tool
2. Digital twin target development environment

Stakeholders:

1. Consultant
2. Capability owner

4.5 Methods Components: Pattern Identification

Method component name: Pattern Identification

Usage objective: Patterns are reusable knowledge on providing secure and resilient services. The method component concerns identification of new patterns according to analysis of several capability models and application cases (Figure 9).

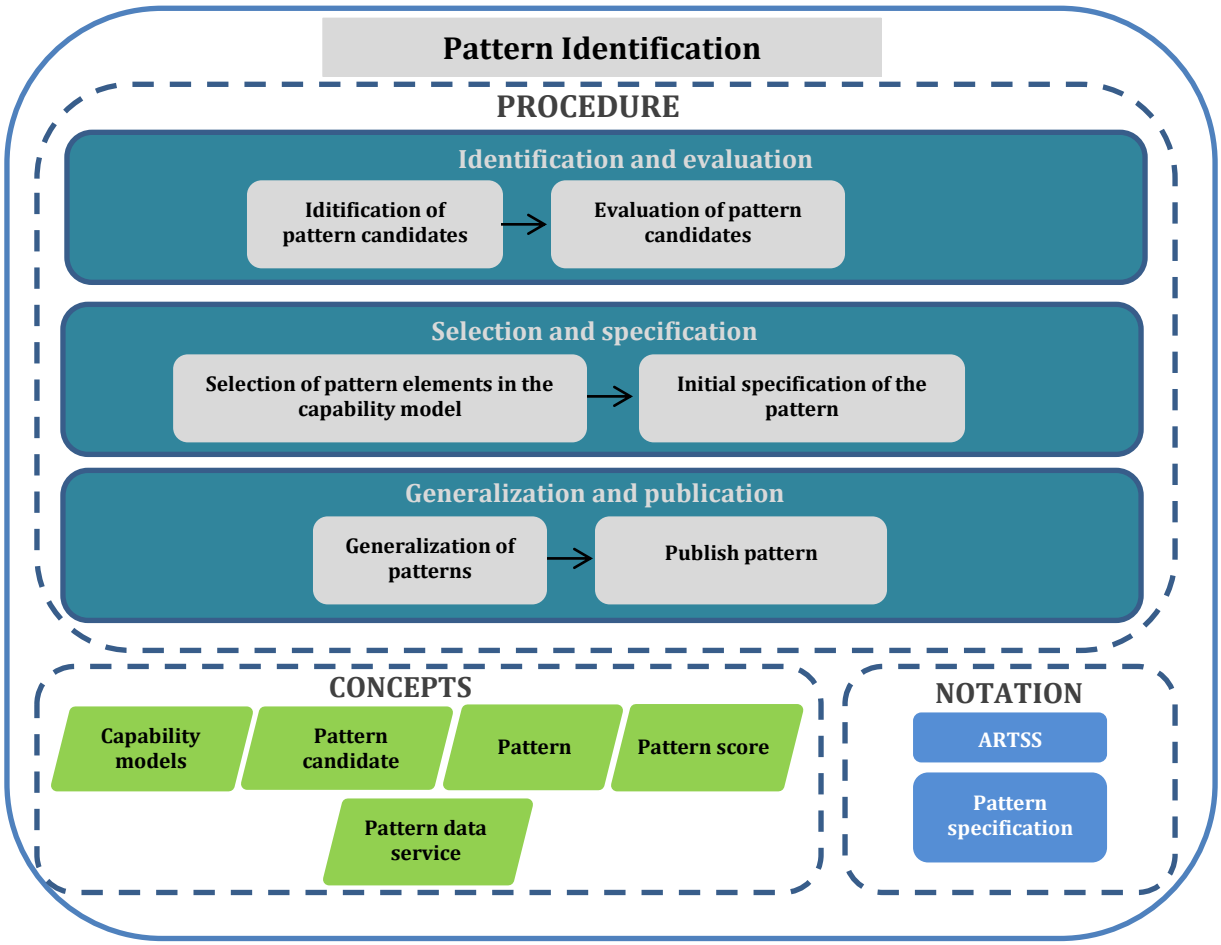


Figure 9 . Pattern identification method component.

Main concepts:

1. Capability models – capability models used to identify pattern candidates;

2. Pattern candidate – a potentially reusable solution;
3. Pattern – reusable knowledge about providing of secure and resilient services;
4. Pattern score – evaluation of utility of the pattern by its users;
5. Pattern data service - pattern specification in JSON format available as a RETS service

Notation:

1. ARTSS notation
2. Pattern specification notation

Process:

Input data:

1. Capability model

Activities:

1. Identification of pattern candidates – capability models are reviewed to identify reoccurring elements or sets of elements, which could be reusable in capability design. Potentially reusable elements in the case of secure and resilient services are:
    a. Adjustments;
    b. KPI characteristic in monitoring and evaluation of security and resilience concerns;
    c. Calculation of context elements;
    d. Measurable properties characterizing security and resilience concerns including non-traditional data sources.
2. Evaluation of pattern candidates – if a pattern candidate reoccurs in several capability models or is deemed as reusable by an expert than the pattern is added to the pattern repository for further specification. Initially, the patterns are added as unpublished items.
3. Selection of pattern elements in the capability model – the pattern specification includes fragments of the capability model what includes reusable elements. It includes the reusable elements as well as related elements from the capability model, e.g., KPI related to the adjustment or measurable properties related to the context element.
4. Initial specification of the pattern – patterns specification notation is used:
    a. Keywords describe the pattern usage area;
    b. Problem – textual description of security and resilience challenges the pattern is intended to deal with;
    c. Goal – describes resilience and security goals the pattern is suitable for. The goals can be represented both textually and graphically;
    d. Performance indicators – measures achievement of the pattern goals. Performance indicators can be represented both textually and graphically;
    e. Context – context defining pattern applicability. Context can be represented both textually and graphically;
    f. Solution – solution to address the defined security and resilience problem to achieve the goals in the specified contextual situation. The solution can be specified both textually and as a diagram.
    g. Input parameters – parameters used to configure the pattern for specific applications.
    h. Guidelines – pattern usage considerations.

5. Generalization of patterns – patterns are reviewed to identify commonalities among the patterns and synchronize specification terms with those included in the ontology of secure and resilient services.
6. Publish pattern – fully specified and approved patterns are published in the pattern repository. The published pattern includes a JSON specification available as a data service.

Results:

1. Short description of pattern candidates.
2. Pattern specification.

Tool Support:

1. ARTSS modeling tool
2. Pattern repository

Stakeholders:

1. Capability owner
2. Sage

4.6 Method Component: Pattern Consumption

Method component name: Pattern consumption (Figure 10)

Application objective: Search of patterns suitable for dealing with resilience and security challenges. The method component is used to 1) create new capabilities models and suitable model elements are needed, e.g., which measurable property to use; 2) a new service requiring resilience and security solutions is being developed; and 3) recommendation on actions to be take in the crisis situations are needed.

The usage of pattern ontology is further elaborated in the "ARTSS ontology and risk management approach" deliverable and the ontology documentation https://imantszarembo.github.io/artss/.

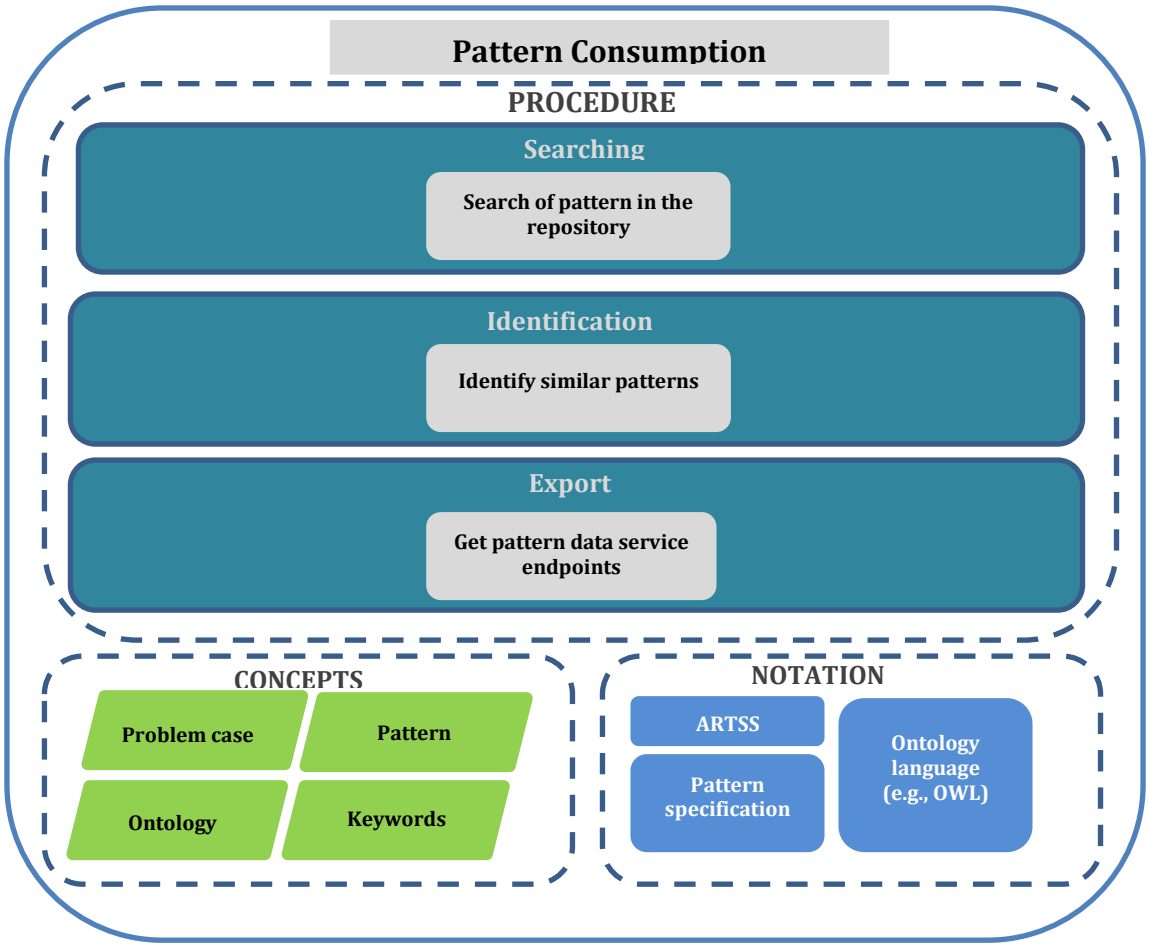


Figure 10 . Pattern usage method component.

Main concepts:

1. Problem case
2. Patterns
3. Keywords
4. Ontology

Notation:

1. ARTSS notation
2. Pattern specification notation
3. Ontology language (e.g.., OWL)

Procedure:

Input data:

1. Keywords describing the problem case
2. Goals and context describing the problem case

Activities:

1. Search of pattern in the repository – search is performed according to keywords, goals or context elements;
2. Identify similar patterns – the ontology is used to reason about the patterns similar those matching the search parameters;
3. Get pattern data service endpoints.

Results:

1. Problem case solutions
2. Pattern data services

Tools:

1. Pattern repository

Participants:

1. Capability owner
2. Service consumer
3. Sage

4.7 Method Component: Capability Ecosystem

The capabilities are continuously developed in the ecosystem by accumulating and formalizing secure service delivery network management knowledge (Figure 11). Knowledge is accumulated during delivery of resilient and secure services including knowledge about suitable adjustments and the best data sources and their providers. This knowledge is represented using patterns and members of the ecosystem provide their feedback on utility of the patterns. The highly valued patterns are used to complement the existing capability models by adding new elements such as adjustment, context elements and measurable properties.

The capability ecosystem method component provides a general overview of dynamic relationships in the capability driven service development and provisioning ecosystem. The capabilities models are used to uncover and to analyze the ecosystem. The capability models published in the pattern repository allow to discover:

1. Mutual interactions among service consumers and providers and data and knowledge flows among the members of the ecosystem;
2. Dependences among the patterns and usage of the patterns in different capability models.

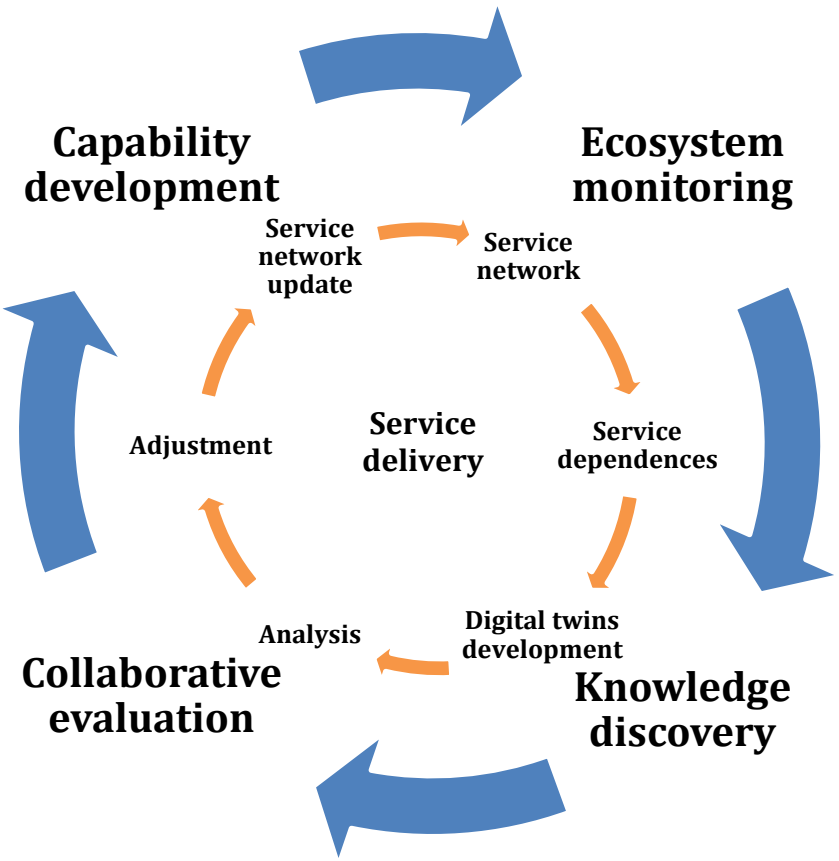


Figure 11 . Capability driven service development and provisioning ecosystem.

4.7.1 Ecosystem Modeling

The ARTSS meta-model is used to model a data ecosystem. In the modeling process, a capability model and an ecosystem view are distinguished. The capability model is created by a modeler who used the ARTSS meta-model and the ecosystem view is generated from the capability model to highlight parties in the ecosystem and their interactions. A fragment of the capability model for a winter road maintenance problem is shown in Fig. 12 (the problem is elaborated in Grabis et al. (2021)). The model provides the capability view of the ecosystem. The focal point of this view are capabilities provided by organizations involved in the ecosystem. In this case, the Road conditions monitoring capability is shown. It concerns organization's ability to determine the current road conditions, for example, bare, partly covered and covered. The model shows the Road conditions monitoring capability provided by the Road monitoring service (among other services not shown in the picture), which in turn invokes the Smart sign management service. Depending on the road monitoring results, appropriate warnings are displayed on smart roadside information boards. The type of warning is determined using the Select warning adjustment, which derives its recommendations according to the Driving conditions context element measured by roadside weather stations. These weather stations provide several measurements including a qualitative evaluation such as snow, icy road, water on icy road, slush on road, freezing rain. The model shows the parties using or providing specific services or assets. The municipality has road monitoring capability while IT company provides monitoring solutions and road management company provides road monitoring tools.

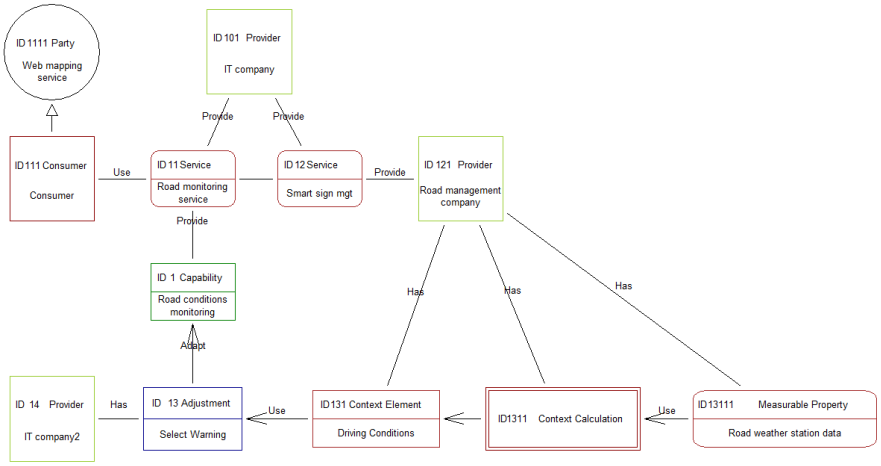


Figure 12 . A fragment of the capability model.

From the ecosystem perspective, the main concerns are interactions among the parties and their roles in the ecosystem. The capability model is processed and analyzed to obtain different views and properties of the ecosystem. For the purposes of the analysis, the ARTSS model is perceived as a property graph (Francis et al, 2018). The property graph of the aforementioned capability model is shown in Fig. 13. It consists of nodes *n* (set of nodes $\mathcal{N}$) and relationships (edges) *r* (set of relationships $\mathcal{R}$). Node labels $\mathcal{L}$ shown in the box attached to the node correspond to the concepts of the ARTSS meta-model:

$$\mathcal{L} = \{\text{Capability, Measurbale Property, Context Element, Adjustment, ...}\}$$

and the relationship types $\mathcal{T}$ correspond to the associations in the ARTSS meta-model

$$\mathcal{T} = \{\text{Evolve, Has, Provide, ...}\}.$$

The nodes have properties represented as key-value pairs and a set of the property keys is denotes as $\mathcal{K}$. Every node has the name property and other properties as specified in the meta-model. A set of the property values is denoted as $\mathcal{V}$. There is function λ that maps nodes to their label (i.e., determines the type of the node). The function τ maps relationships to their types.

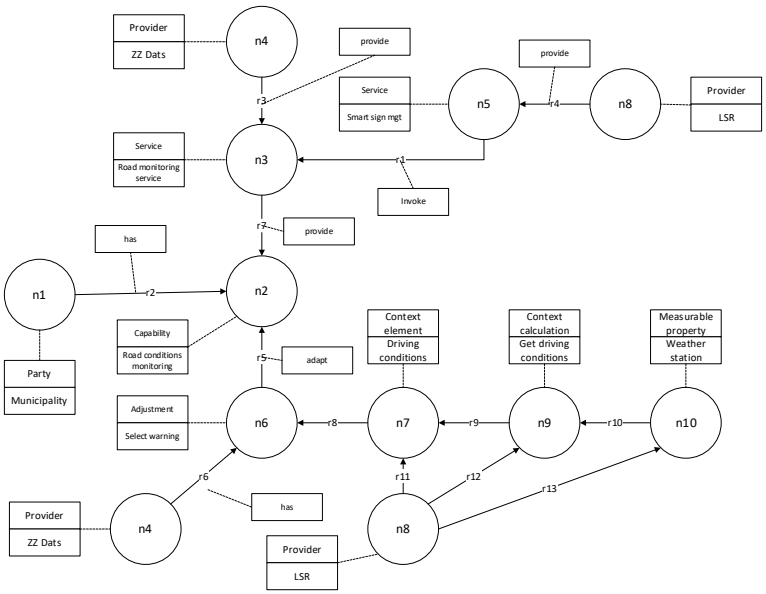


Figure 13 . A fragment of the capability model represented as property graph.

A chain of relationships among the classes forms a path from one object to another. The path consists of nodes and relationships to be traversed to reach one node from another, and it stands as a proxy for describing interaction in the capability model. There are several sets of paths of the interest in the ecosystem model:

$\mathcal{P}1$. Measurable property to Capability;
$\mathcal{P}2$. Consumer to Provider;
$\mathcal{P}3$. Party to Consumer;
$\mathcal{P}4$. Asset to Consumer;
$\mathcal{P}5$. Consumer to Capability.

These paths are used in the analysis of the ecosystem. The set of paths $\mathcal{P}_1$ formally is evaluate as

$$\mathcal{P}_1 = \left\{ n_i r_i ... r_j n_j \mid \lambda(n_i) = \text{MeasurableProperty}, \lambda(n_j) = \text{Capability} \right\},$$

where $n_i r_i$. refers to the node and relationship attachment and three dots denote that any nodes and relationships can be traversed from the starting node to the end node. The $\mathcal{P}_2$ path is determined as

$$\mathcal{P}_2 = \left\{ n_i r_i ... r_j n_j \mid \lambda(n_i) = \text{Consumer}, \lambda(n_j) = \text{Provider} \right\}.$$

The other sets of paths are determined in a similar manner.

The ecosystem view is derived by inferring interactions among the parties in the ecosystem from the capability based-model. The ecosystem view is a graph consisting of ecosystem parties as nodes and interactions among the parties as relationships. In the graph, the interactions are represented as ecosystem relationship types $\mathcal{T}_E$ There is an open set of interactions of the interest to analysts of the ecosystem. The following interactions are currently considered:

I1. Measurable Property provider – a provider of measurable properties for the capability delivery. The path traversed is Party > Capability > Context Element < Measurable Property < Provider;
I2. Adjustment provider – a provider of adjustments to the capability party. The path traversed is Party > Capability < Adjustment < Provider;
I3. Service provider – a provider of services to the capability party. The path traversed is Party > Capability < Service < Provider;
I4. Capability enabler – a provider that makes available assets need by a party to deliver its capability. The path traversed is Party > Capability < Asset < Provider;
I5. Service consumer – links service providers and consumers. The path traversed is Consumer > Service < Provider;
I6. Joint service – a service, which requires collaboration of multiple providers. The path traversed is Provider > Service < Provider;
I7. Shared goals – goals common to multiple parties in the ecosystem. The path traversed is Party > Capability > Goal < Capability < Party;
I8. Shared capability – a capability possessed by multiple parties in the ecosystem. The path traversed is Party > Capability < Party;
I9. External adjustment – the adjustment provided to ecosystem parties not directly involved in capability delivery. The path traversed is Provider > Adjustment > Service < Consumer.

Formally, the Measurable Property Provider interaction I1 is identified by applying the following rule:

$$\mathcal{R} \cup \{r_i\}, \mathcal{T}(r_i) = \text{I2}, \text{src}(r_i) = n_j, \text{tgt}(r_i) = n_k, \text{path}(n_j, n_k) \in \mathcal{P}_2 \quad ,$$

where scr() is a function that determines the source of the relationship, tgt() is a function that determines the target of the relationship and path() is a function that determines the path between two nodes. The rule creates a I1 type of relationship between two parties if a measurable property by a provider is used in a capability by a party. Other interactions are defined in a similar manner.

The rules defined are applied to the capability model and the ecosystem view is created (Fig. 14). It clearly identifies all the parties involved in the data ecosystem and their interactions. For example, party P1 has a Measurable property need by P2. The ecosystem view is further used in the ecosystem model analysis.

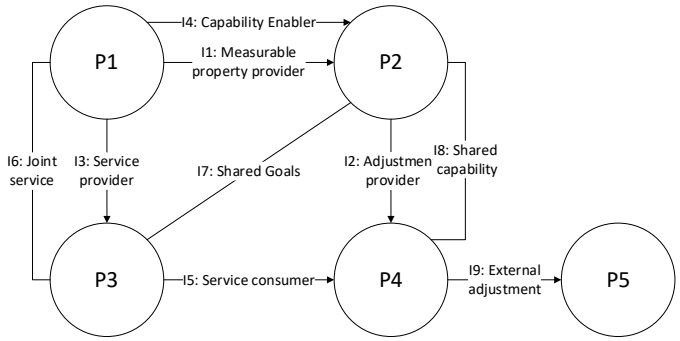


Figure 14 . The ecosystem view of the ARTSS capability model.

4.7.2 Model Analysis

The model analysis is performed to comprehend interactions in the ecosystem and to evaluate its resilience. The model analysis is performed in several stages (Fig. 15). The capability model is created using the ARTSS extension of the CDD method and it shows ownership and service provisioning relationships of assets and services, respectively. The ecosystem view is derived to highlight interactions among the parties in the ecosystem. The capability model and its ecosystem view are analyzed to evaluate properties of the data ecosystem. The ecosystem model can be used to create capability models and to setup capability delivery solutions for individual parties in the ecosystem, though this aspect is aspect is beyond the scope of this paper.

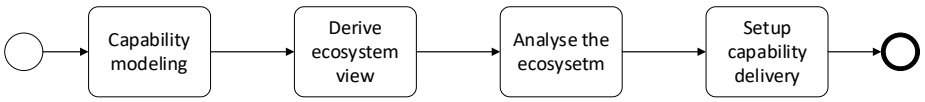


Figure 15 . The ARTSS model analysis activities

The ecosystem model is analyzed to evaluate resilience and other properties of the ecosystem. From the resilience perspective, there is a specific concern on the impact of loosing some of the services or assets on the overall resilience of the ecosystem and capability delivery. Three types of measure are used to evaluate the resilience:

1. The impact of node deletion;
2. The degree of substitution;
3. The centrality of parties.

The node deletion (i.e., an asset or service becomes unavailable) affects another node of if there is a path between the nodes what is determined by the indicator $\pi_{n_i n_j}$

$$\pi_{n_i n_i} = \begin{cases} 1, \text{ if path}(n_i, n_j) \in \mathcal{P} \\ 0, \text{otherwise} \end{cases}.$$

For example, if the Weather station data measurable property is lost then assets like Select warning adjustment are affected. The overall ecosystem resilience is evaluated by the number of consumers (*CNSA*) and capabilities (*CPBA*) affected due the deletion of selected nodes and these measures are calculated using equation (1) and (2), respectively:

$$CNSA = \sum_i \sum_j \pi_{n_i n_j}, \forall \lambda(n_i) \in \{Service, Asset\}, \lambda(n_j) \in \{Consumer\} \quad (1)$$

$$CPBA = \sum_i \sum_j \pi_{n_i n_j}, \forall \lambda(n_i) \in \{Service, Asset\}, \lambda(n_j) \in \{Capability\} \quad (2)$$

The resilience of specific service or capability is evaluated as a count of services or assets disabled due to the node deletion. The former (service count) is used during the capability delivery. The latter (asset count) is used during the capability design if specific implementations of assets are not known.

The degree of substitution (*DS*) is specified as a number of providers for an asset or service:

$$DS(i) = \sum_j \sigma_{n_i n_j}, \forall \lambda(n_{ij}) \in \{\text{Provider}\},$$

where $\lambda(n_i) \in \{Service, Asset\}$ and $\sigma_{n_i n_i} = \begin{cases} 1, \text{ if } \left|\text{path}(n_i, n_j)\right| = 1 \\ 0, \text{otherwise} \end{cases}$. The expression specifies that for given service or asset all dicretly connected providers are counted.

The centrality of parties is determined by its out-going degree (*NC*):

$$NC(i) = \sum_j \left(\text{src}(r_j) = i\right), \forall \tau(r_j) \in \mathcal{T}_E,$$

where $\lambda(n_i) \in \{\text{Party,Provider,Consumer}\}$. The expression specifies that for a given party all originating associations of type $\mathcal{T}_E$ are counted.

The centrality indicates parties potentially having the most significant impact on the ecosystem. It is calculated for specific types of interactions.

5 Tools

Development of secure and resilient services from the capability-oriented perspective is supported by the ARTSS toolset[1]:

1. ARTSS modeling tool – the modeling tool is used to formally specify capabilities and services. The modeling results are a capability model as a diagram, which is used of analysis purposes, as well as a capability model in machine readable format, which is used development of digital twin and resilient and secure services
2. Digital twin development platform – development environment and set of libraries used to create digital twins in the basis of the capability model;
3. Pattern repository – knowledge base where patterns on development and delivery of resilient and secure services are store in a structured manner. The repository provides pattern management, search and evaluation functionality;
4. E-learning solution – modularized e-learning courses providing training materials on secure and resilient services.

[1] See D3.1 for access information

6 Example

In the project the ARTSS method is used in several use cases. The foundational services use case investigates resilience and security aspects in computer networks. This section provides a brief overview of the use case and capability modeling and more information can be found in the use case documentation deliverable (ref#).

6.1 Secure Foundational Services

The foundational ICT services are those enabling operations of other digital services and secure networking in particular (Figure 16). With much of the economy and learning happening on-line, malicious activities in a computer network should be identified as soon as possible.

A campus area network (CAN) consists of multiple interconnected local area net-works (LAN) in a limited geographical area. It is often characterized by combination of different modes of management and access and security control in particular. The example explores the case of CAN governance at a large higher education institution. A large higher education institution operates CAN. The network is highly heterogeneous consisting of multiple LAN with different security settings and governance modes. One part of the network is governed by a centralized network management system while other parts are not. There is a variety of devices connecting to the net-work including private computers and specialized devices. There are locations and situations permitting access without authentication. As a consequence of the COVID-19 crisis, studies are taking place on-line and employees work remotely what puts strains on network performance and creates additional security concerns.

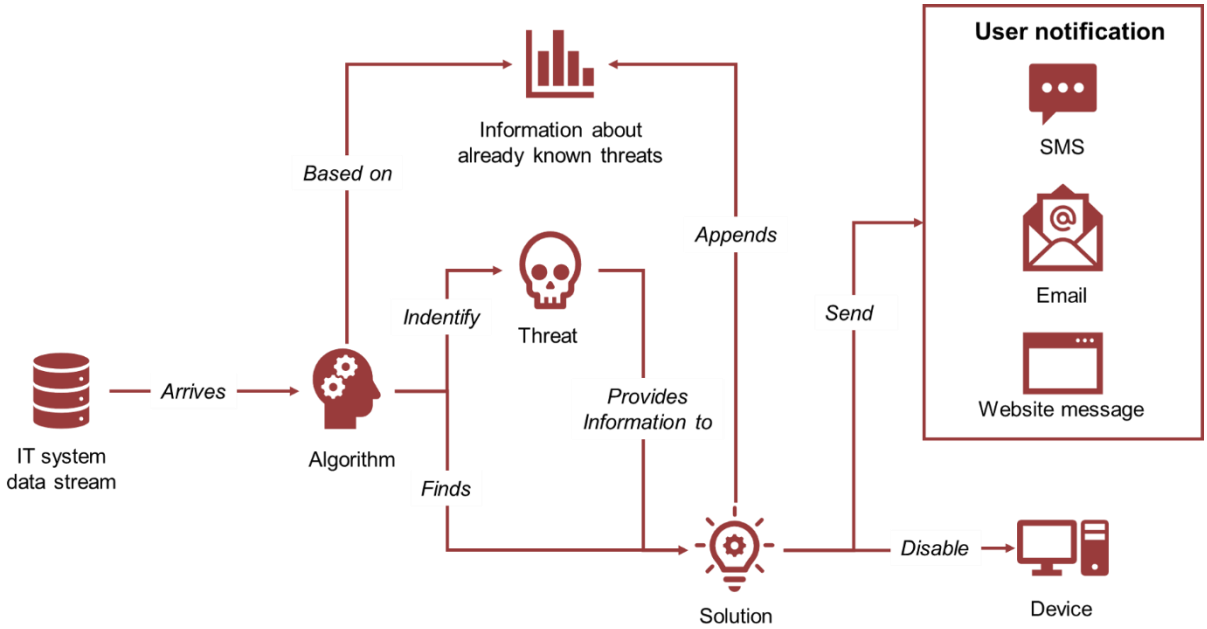


Figure 16 . The overview of malicious activity identification in computer networks

6.2 Capability model

In order to ensure continuous operations of the higher education institution, the Secure campus network governance capability is deployed. Following the CDD methodology, a capability model is developed (Fig. 13). The capability fulfills To provide secure IT governance goal supported by the goals To provide high connectivity and To prevent security incidents. The former specifically targets an ability of all users to access network, especially, during live events. The latter concerns security. That is also supported by the goal To minimize warning to CERT, what is an external network security monitoring organization.

The capability delivery is affected by relevant context elements. The Device threat level indicates whether a network connected device is potentially infected what is evaluated using multiple measurable properties. The evaluation is not always 100% accurate because of obfuscation. The user threat level depends on availability of the user identity information and user's previously recorded behavior. The Urgency context element characterizes the need to resolve security incidents as quickly as possible. It is assumed that the urgency increases if there are many active users and external data indicates intensification of security incidents. The context elements are evaluated using the specified measurable properties. It is important to note that these are the elements considered by the particular institution and other institutions would design the capability and choose elements according to their needs. The capability is implemented using two primary services, namely, the Malicious activity identification service and the Incident resolution service. The former uses a number of techniques to identify infected nodes in the network [14] and the latter provides various means for informing users on activities required to resolve the incident.

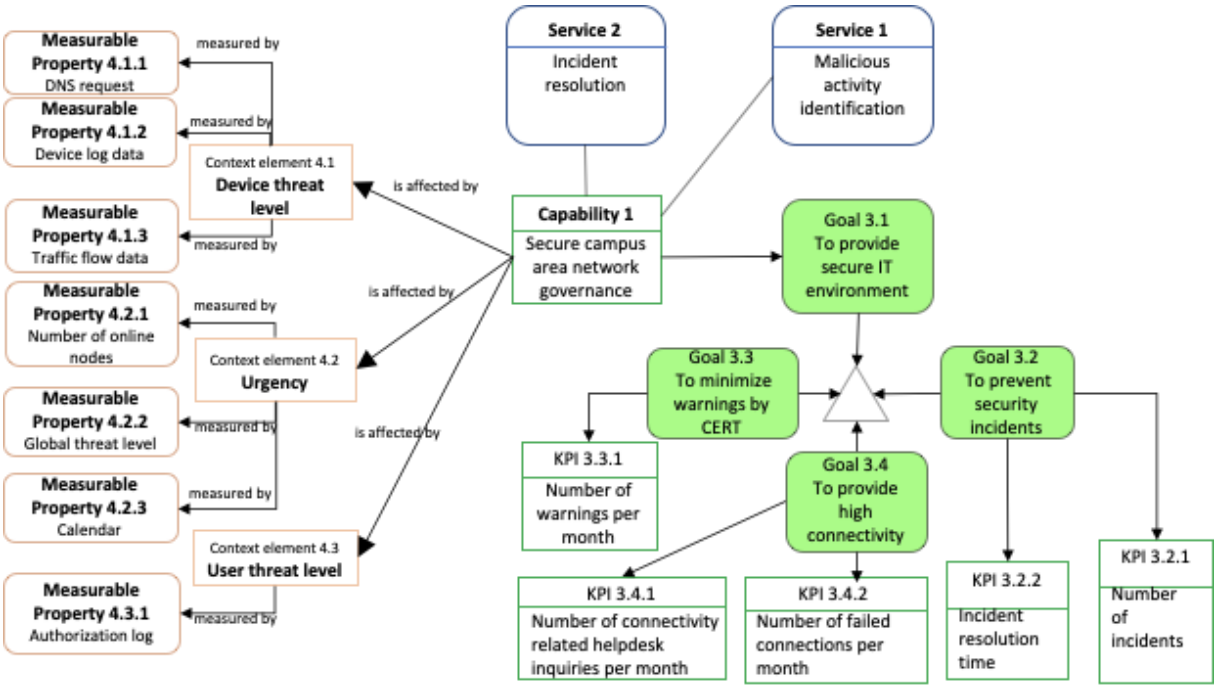


Figure 17 . Secure network governance capability model: goals and context

In order to identify means for ensuring security and resilience, the capability model is further refined (Fig. 14). Potential values of the context elements are identified, and context ranges are specified. These values guide the capability delivery in response to changes in the context. They are defined as categorical variables and the categories are defined by the modelers and using data analysis. The Device threat level assumes values {None, Possible Low, Possible High, Definite low, Definite high}. The low values indicate that a threat identified does not

create immediate harm while the high values suggest that the node could affect operation of the whole network. The latter case requires an immediate action to resolve the incident. The possible values suggest detection of suspicious activities though they cannot be classified as malicious. The definite values indicate that the node is definitely infected. The Urgency has categories {Low, Medium, High}. The high level is assigned if there are many active nodes during crucial operation hours. The User threat level is determined according to the previously observed behavior of the respective user and it is set to Unknown if the user cannot be identified. Adjustments to adapt the capability are identified. The Select response adjustment determines, which options provide by the Incident resolution service should be invoke depending on the context and KPI. It is identified that two types of responses could be used: 1) Notify user; and 2) Disconnect device. These responses are treated as sub-capabilities because the institution should possess abilities to carry out the response mechanisms and they are potentially reusable for other organizations.

The Select response adjustment invokes either the Notify user capability or the Disconnect device capability. The decision is made according to a decision-table specifying the selection outcome according to the context values. The Disconnect device capability is deployed if User threat level is Unknown and Urgency is Low or Medium because disconnecting one node could affect other nodes urgently requiring the network. The Notify user capability uses the Select notification type adjustment, which select among posting a message into user profile, sending an e-mail message or sending and SMS message. The latter has been observed as the most efficient notification type (i.e., user response time is the shortest) though it is also the most intrusive and its over-usage could decrease it efficiency. Therefore, it is preserved for case when either of the context elements assume value High. Selection of suitable response and notification means directly affects resilience of the system to respond to security incidents and to ensure ongoing network operations.

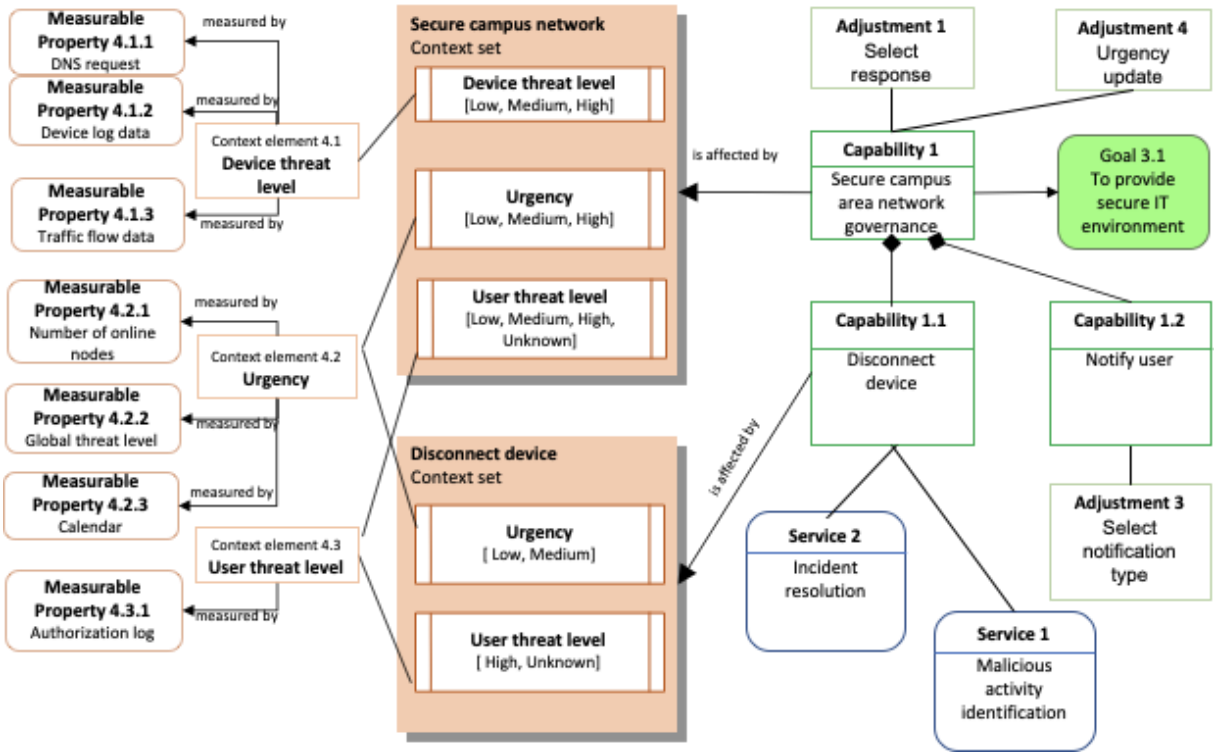


Figure 18 The Secure campus area network governance capability and its sub-capabilities

The Urgency update adjustment is also introduced (Fig. 15). This adjustment dynamically changes classification of the incidents according to their urgency. If there are too many

incidents and their resolution time is too long then more incidents are classified as urgent requiring efficient resolution mechanisms.

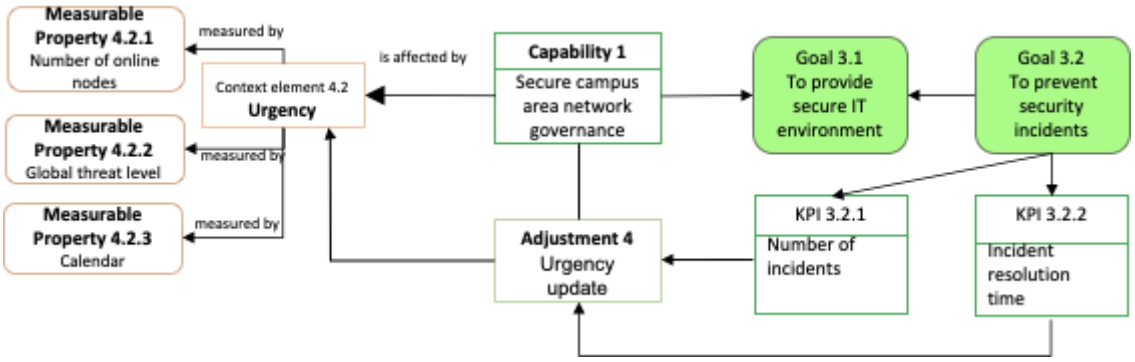


Figure 19 . The Urgence update adjustment

References

Athinaiou, M., Mouratidis, H., Fotis, T., Pavlidis, M., Panaousis, E.: Towards the definition of a security incident response modelling language. International Conference on Trust and Privacy in Digital Business, TrustBus 2018, pp. 198-212 (2018).

Bodeau D, Graubart R.: Cyber resiliency design principles. United States: The MITRE Corporation; 2017. Jan, pp. 1–90. Technical report, Report No: 17-0103 (2017).

Byers, D., Shahmehri, N.: Unified modeling of attacks, vulnerabilities and security activities", Proceedings - International Conference on Software Engineering, pp. 36. (2010)

De Reuver, M., Sørensen, C., Basole, R. C. : The Digital Platform: A Research Agenda. Journal of Information Technology 33(2), 124–135 (2018).

Eckhart, M., Ekelhart, A., Weippl, E.: Enhancing Cyber Situational Awareness for Cyber-Physical Systems through Digital Twins. IEEE International Conference on Emerging Technologies and Factory Automation, 1222 (2019).

Elahi, G., Yu, E.: Modeling and analysis of security trade-offs - A goal oriented approach. Data and Knowledge Engineering 68(7), 579-598 (2009).

Fiksel, J.: Designing Resilient, Sustainable Systems. Environmental Science & Technology. 37(23), 5330–5339 (2003).

Francis, N. et al.: Cypher : An Evolving Query Language for Property Graphs. SIGMOD '18: Proceedings of the 2018 International Conference on Management of Data, May 2018, pp. 1433–1445 (2018).

Grabis J., Kampars J. (2018) Adjustment of Capabilities: How to Add Dynamics. In: Sandkuhl K., Stirna J. (eds) Capability Management in Digital Enterprises. Springer, Cham. https://doi.org/10.1007/978-3-319-90424-5_8

Grabis, J., Stirna, J., Deksne, L., Roponena, E. (2021). A Capability Based Method for Modeling Resilient Data Ecosystems. Domain-Specific Conceptual Modelling: Concepts, Methods and Tools, Springer (submitted)

Grabis, J., Stirna, J., Zdravkovic, J. (2020) A Capability Based Method for Development of Resilient Digital Services, Selected Papers of ICEIS 2020, Springer LNBIP (in press)

Goldkuhl G., Lind M., Seigerroth U.: Method integration: the need for a learning perspective. IEE Proceedings Software 145(4), 113-118 (1998).

Haque, Md A., Kamdem De Teyou, G., Shetty, S. and Krishnappa, B.: Cyber Resilience Framework for Industrial Control Systems: Concepts, Metrics, and Insights. In Proc. of IEEE International Conference on Intelligence and Security Informatics Conference, ISI, IEEE, DOI: 10.1109/ISI.2018.8587398 (2018)

Henkel M., Zdravkovic J., Valverde F., Pastor O. (2018) Capability Design with CDD. In: Sandkuhl K., Stirna J. (eds) Capability Management in Digital Enterprises. Springer, Cham. https://doi.org/10.1007/978-3-319-90424-5_6

Kampars, J., Zdravkovic, J., Stirna, J., Grabis, J.: Extending organizational capabilities with Open Data to support sustainable and dynamic business ecosystems, Software and Systems Modeling 19, 371–398 (2020).

Koç H., Sandkuhl K. (2018) Context Modelling in Capability Management. In: Sandkuhl K., Stirna J. (eds) Capability Management in Digital Enterprises. Springer, Cham. https://doi.org/10.1007/978-3-319-90424-5_7

Korpela, K., Kuusiholma, U., Taipale, O., Hallikas, J.: A framework for exploring digital business ecosystems. In: 46th Annual Hawaii International Conference on System Sciences HICSS 2013, pp. 3838–3847. Institute of Electrical and Electronics Engineers Inc. (2013).

Kritzinger, W., Karner, M., Traar, G., Henjes, J., Sihn, W.: Digital Twin in manufacturing: A categorical literature review and classification, IFAC-PapersOnLine 51(11), pp. 1016-1022 (2018).

Kampars, J., Zdravkovic, J., Stirna, J., Grabis, J.: Extending organizational capabilities with Open Data to support sustainable and dynamic business ecosystems, Software and Systems Modeling 19, 371–398 (2020).

Mouratidis, H., Argyropoulos, N., Shei, S.: Security requirements engineering for cloud computing: The secure Tropos approach. In: Domain-Specific Conceptual Modeling: Concepts, Methods and Tools, pp. 357-380, Springer, Cham (2016).

Ross, R. Pillitteri, V., Graubart, R., Bodeau, B., McQuaid, R. Developing. Cyber Resilient Systems: A Systems Security Engineering Approach. SP 800-160 Vol.2, https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-160v2.pdf Last accessed: 28/05/20 (2019)

Sandkuhl, K., Stirna, J.: Capability Management in Digital Enterprises. Springer, Cham (2018).