Title:	*Trustworthiness mechanisms specification*
Authors:	*Erkuden Rios (TECNALIA), Eider Iturbe (TECNALIA), Angel Rego (TECNALIA), Anne Gallon (EVIDIAN), Thierry Winter (EVIDIAN), Hui Song (SINTEF)*
Editor:	*Erkuden Rios (TECNALIA)*
Reviewers:	*Arnor Solberg(TellU), Uģis Grīnbergs (BOSC), Nicolas Ferry (SINTEF)*
Identifier:	*Deliverable # D4.1*
Nature:	*Report*
Date:	*31 October 2018*
Status:	*Final*
Diss.	*level:Public*

Executive Summary

This deliverable provides an overview of the state-of-the-art mechanisms for trustworthiness of Smart IoT Systems, particularly security, privacy and resilience mechanisms. In addition, the report describes the intended support and advance over state-of-the-art that ENACT solution will offer to these aspects in both Development and Operation phases.

The research leading to these results has received funding from the European Community's H2020 Programme under grant agreement n° 780351 (ENACT).

Members of the ENACT consortium:

Stiftelsen Sintef	Norway
CA Technologies Development Spain S.A.	Spain
EVIDIAN SA	France
INDRA Sistemas SA	Spain
FundacionTecnalia Research & Innovation	Spain
TellU AS	Norway
Centre National de la Recherche Scientifique	France
Universitaet Duisburg-Essen	Germany
Istituto per Servizi di Ricovero e Assistenza agli Anziani	Italy
Baltic Open Solution Center	Latvia
Elektronikas un Datorzinatnu Instituts	Latvia

Revision history

Date	Version	Author	Comments
05-04-2018	V0.1	Erkuden Rios	TOC proposed
04-05-2018	V0.2	Erkuden Rios	TOC updated and agreed
11-05-2018	V0.2.1	Anne Gallon	SOTA of AC
12-05-2018	V0.2.2	Hui Song	SOTA of Resilience
01-06-2018	V0.3	Erkuden Rios	SOTA of Requirements
29-06-2018	V0.3.1	Anne Gallon	AC mechanism description
11-07-2018	V0.3.2	Hui Song	Diversity mechanisms description
19-09-2018	V0.4	Erkuden Rios	SOTA of IoT security and privacy requirements specification and monitoring, SOTA of IoT threats, SMOOL security mechanism description.
09-10-2018	V0.4.1	Erkuden Rios	WP1 Requirement Analysis documented
21-10-2018	V0.5	Erkuden Rios	Document reshaped. Full description of Enablers included. Version for Internal review.
30-10-2018	V0.6	Erkuden Rios	Revised version addressing Internal review feedback.
31-10-2018	V1.0	Erkuden Rios	Final version for submission.

Contents

1 Introduction

1.1 Context and objectives

Large-scale distributed Internet of Things (IoT) systems pose major challenges with respect to how to address their security and privacy concerns efficiently. In particular, security-by-design and privacy-by-design methods and tools are required to address a holistic design embracing security and privacy aspects at the different system layers. Furthermore, automated solutions for run-time operations are required in order to ensure timely reaction to privacy and security incidents, occurred accidentally or caused by attackers.

The WP4 in ENACT aims at developing methods and tools supporting the security, privacy and resilience of Smart IoT Systems (SIS) throughout the DevOps process cycle (see Figure 1). Smart IoT Systems in ENACT are next generation IoT systems which need to perform distributed processing and coordinated behaviour across IoT, edge and cloud infrastructures, manage the closed loop from sensing to actuation, and cope with vast heterogeneity, scalability and dynamicity of IoT devices and their environments.

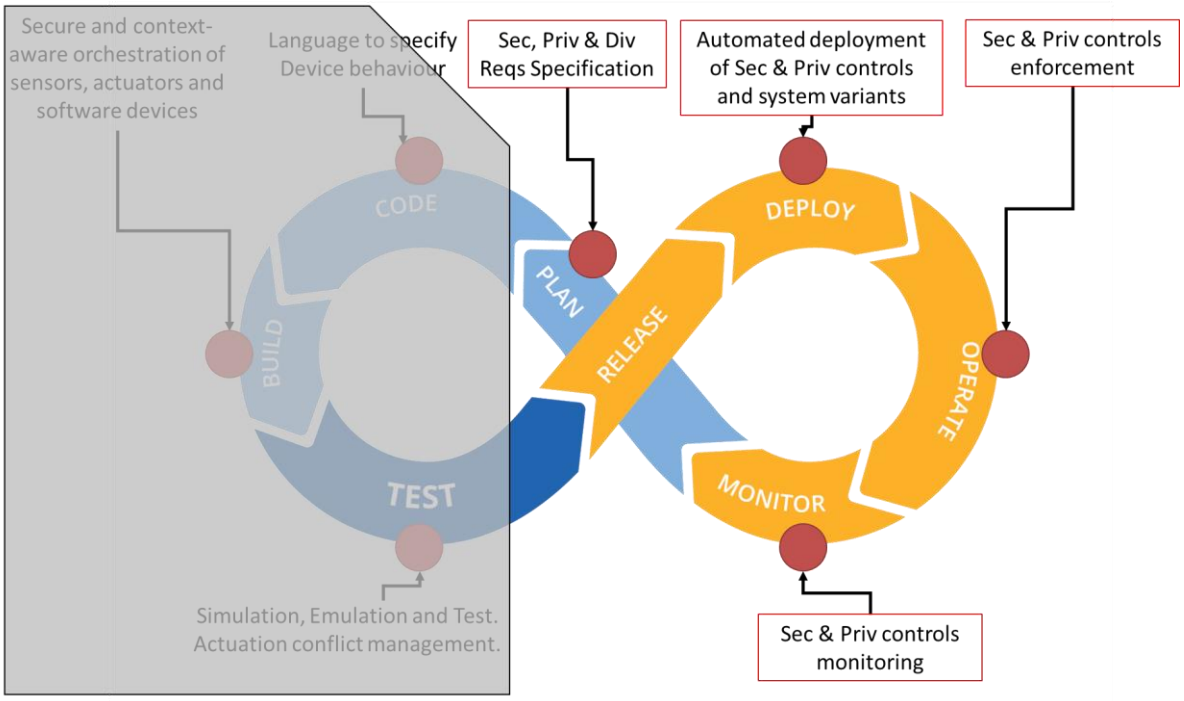

Figure 1 – ENACT WP4 focus within DevOps cycle

As explained in D2.1, within ENACT we define "***Trustworthiness"*** as the capability to *"preserve security, privacy, safety, reliability, and resilience of SIS"*.

From all these, the WP4 in ENACT deals with supporting the following SIS capabilities:

- **Security** refers to the preservation of confidentiality, integrity and availability of information [1].
  - **Integrity** is the property of protecting the accuracy and completeness of information [2].
  - **Confidentiality** is the property that information is not made available or disclosed to unauthorized individuals, entities, or processes [2].
  - **Availability** is the property of information being accessible and usable upon demand by an authorized entity [2].

- **Privacy** refers to the protection of personally identifiable information (PII) [3][1]. PII refers to any information that (a) can be used to identify the PII principal to whom such information relates, or (b) is or might be directly or indirectly linked to a PII principal.
- **Resilience** refers to the ability of the SIS to withstand instability, unexpected conditions, and gracefully return to predictable, but possibly degraded, performance [4].

Therefore, within WP4 we will research novel mechanisms related to security and privacy (including access control) as well as resilience of SIS, while safety and reliability aspects are not studied. In terms of resilience we will leverage software diversity and deployment of different system variants with the aim to reduce the exposure of particular faults of the system to potential attackers as well as increase the resilience of the system against external perturbations.

The results of WP4 will be shaped as two main enablers:

- **Robustness and Resilience Enabler:** In order to contribute to SIS trustworthiness this enabler will increase resilience of smart IoT systems by diversifying software. This implies that each instance of a service has a different implementation and it operates differently, still ensuring that its global behaviour is consistent and predictable. The enabler will automate the introduction and management of diversity in smart IoT systems.
- **Security and Privacy Monitoring and Control Enabler:** This enabler includes mechanisms and tools to control the security and privacy behaviour of IoT systems and to early detect anomalies by continuous monitoring of security metrics that will be defined during the project. This includes early reaction models and mechanisms that address adaptation and recovery of the IoT application operation in case of monitored metrics deviation from the expected behaviour. Specific focus will be on the confidentiality and integrity of data and services. The enabler will include an end-to-end Context-Aware Access Control tool for advanced access control and authorization mechanisms tailored to smart IoT systems. Today, no protocol can deliver dynamic authorization based on context for both IT (information technologies) and OT (operational technologies) domains.

The present deliverable D4.1 focuses on the state-of-the-art and use case requirement analysis to derive the security, privacy and resilience mechanisms necessary to support security-by-design, privacy-by-design, resilience-by-design as well as monitoring and operational control of these aspects in SIS.

1.2 Achievements

The following table summarises the achievements of WP4 at the time of delivering D4.1.

Table 1. Achievements of ENACT WP4 at the time of D4.1 delivery.

Objectives	Achievements so far and future work
State-of-the-art on IoT security, privacy and resilience	We conducted an extensive analysis of the state-of-the-art on approaches for security, privacy and resilience of SIS. We specifically focused on five topics: 1. IoT Security and Privacy challenges. 2. IoT Security- and Privacy-by-design. 3. IoT Security and Privacy assurance.

[1] International Organization for Standardization. ISO/IEC 29100:2011(E), Information technology – Security techniques – Privacy framework, 2011.

	4. Software diversity as resilience mechanism in SIS.
Security, and privacy-aware design and orchestration of IoT systems i) The languages and formalisms to enable the specification of the security and privacy requirements of smart IoT applications as part of the overall design, including the corresponding security and privacy metrics and probes allowing appropriate monitoring.	Analysis of possible mechanisms and tools related to the expression and inclusion of security and privacy-intelligence in smart IoT systems design. Initial design of monitoring tool ready. Support to security and privacy specification at Orchestration is pending.
ii) Risk model characterizing potential security and privacy risks, considering both the characteristics of infrastructure devices and requirements of the smart IoT application (this will be integrated with WP2 risk driven orchestration and the decision support system for selection of the devices).	Addressed in D2.1
iii) Metrics of software diversity of individual services and the whole system.	On-going research.
Robustness, security and privacy enforcement in smart IoT systems	Initial design of context-based authentication and authorisation of devices and services. Initial design of diversity mechanisms to diversify IoT services, i.e., to automatically generate diverse versions of IoT services from the same ThingML model. In addition, initial research was done on IoT architecture diversity mechanisms. Initial design of IoT Platform level security and privacy mechanisms. In the future possible reaction models and mechanisms will be also defined to address the adaptation and recovery of the IoT application operation in case of monitored metrics deviation from the normal (risk under control) behaviour.
Security and privacy monitoring of smart IoT systems	Initial mechanisms and tools for controlling the security and privacy behaviour of IoT application and early detect anomalies by continuously monitoring.

1.3 Structure of the document

After the introductory section, the reminder of the document is structured as follows.

In Section 2, the Subsection 2.1 describes the main obstacles for a holistic approach to SIS security and privacy. Then, the state-of-the-art in IoT security, privacy and resilience are presented in Subsection 2.2, 2.3 and 2.4 respectively.

Section 3 analyses the requirements of ENACT use case with respect to SIS security, privacy and resilience and explains how it is planned to tackle them in the corresponding enablers of ENACT framework.

Section 4 summarises the IoT Trustworthiness architecture and future tool support in ENACT and describes how the different tools will work together within ENACT framework.

Section 5 describes the different mechanisms being developed as part of ENACT solution to support SIS developers in creating trustworthy SIS.

Section 6 describes the initial design of the operational mechanisms that will be offered by ENACT to SIS operators in order they can effectively detect cyber incidents, anomalies, and attacks and early react to them.

1.4 Acronyms and abbreviations

CAAC	Context-aware Access Control	IDM	Identity Management
CPIM	Cloud Provider Independent Model	IoT	Internet of Things
CPSM	Cloud Provider Specific Model	IP	Internet Protocol
CSP	Cloud Service Provider	SLA	Service Level Agreement
DoS	Denial of Service	SLO	Service Level Objectives
GDPR	EU General Data Protection Regulation	SIS	Smart IoT System

2 State of the art in IoT Security, Privacy and Resilience

Smart IoT system security, privacy and resilience are wide fields of research including mechanisms and solutions applicable in the different layers of the IoT system. The mechanisms range from formalisms and models for specification of security, privacy and resilience requirements in the system design to techniques and methods for requirements fulfilment assurance at system run-time.

This section examines the current state of the art of security, privacy and resilience mechanisms of IoT systems from the perspective of their possible inclusion on ENACT solution. First, in Section 2.1, we introduce the major challenges of IoT systems with respect to IoT security and privacy. Second, in Section 2.2 the state of the art in security-by-design and privacy-by-design mechanisms relevant for ENACT is described. Third, in Section 2.3 the state of the art of run-time security and privacy mechanisms is studied with emphasis on access control solutions. Finally, Section 2.4 analyses the state of the art of both design time and run-time resilience techniques.

2.1 IoT Security and privacy challenges

Smart IoT Systems (SIS) are complex and dynamic systems which require the management of distributed and heterogeneous devices, technologies, services and environments. This heterogeneity implies working with different underlying networks (e.g., wired, wireless, cellular) and supporting different communication protocols (e.g., RFID ISO/IEV 18000, IEEE 802.15.4, ZigBee, Wireless HART, WiFi IEEE 802.11 a/b/g/n, WiMax IEEE 802.16 a/d/e/m, Ethernet IEEE 802.3 u/z, GPRS) and modes (e.g., access point-based, p2p mode) in order to manage massive device data transmission [5]. Due to IoT characteristics, it is challenging to ensure security in terms of identification and authentication, confidentiality, integrity, authorization, availability and privacy, while scalability, high capacity and availability must be guaranteed, in real-time the most of the times. The security requirements need to be addressed by implementing the existing security modes of the communication protocols themselves and by deploying the necessary security mechanisms for data protection at rest as well.

There is no standard architecture for representing the IoT. Nevertheless, there is well-known three-layer architecture that consists of the perception layer, the network layer and the application layer [6] [7]. Security must be ensured at all layers and security of the IoT environment as a whole needs to be addressed as well.

Figure 2 shows the IoT layers mapped to the three different phases that take place into the IoT environments: (i) collection phase, (ii) transmission phase, and (iii) process, management and utilization phase [5]. The horizontal representation of IoT applications illustrates how these applications do not work in isolation but share devices, networks and infrastructure elements, moreover, there is a common service platform that is in charge of managing and controlling them [8].

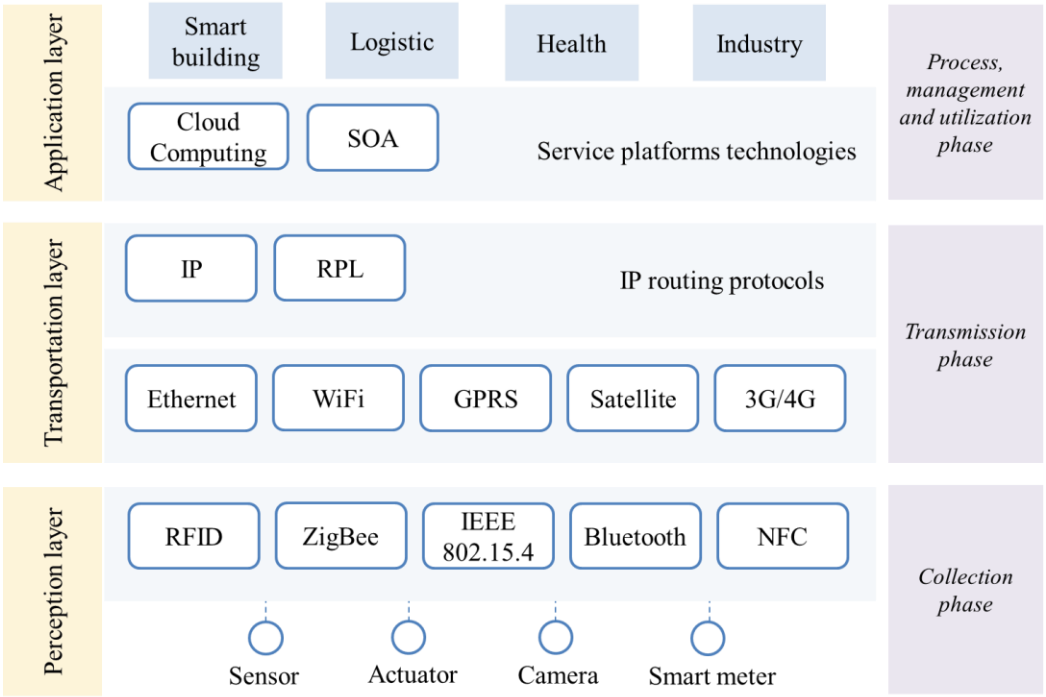

Figure 2 – IoT technologies and protocols stack

The collection phase, which executes at the **perception layer**, refers to procedures for collecting real-time data from the physical environment. Technologies used for the data collection are mainly: RFID ISO/IEV 18000, Wireless Sensor Networks (WSN) technologies such as IEEE 802.15.4, ZigBee, Wireless Highway Addressable Remote Transducer (HART) and NFC. Regarding WSNs, the Low-power wireless personal area networks (LoWPAN) protocol 802.15.4 covers the low-energy communications requirements of IoT systems.

During the transmission phase the collected data is delivered through the network layer to the service platforms and servers at the application layer. Different technologies may be used at **transportation layer** for that purpose such as Ethernet, WiFi, and GPRS. The communication protocol stack for the IoT already supports security. The IEEE 802.15.4 standard defines different security modes by using symmetric cryptography, which assures data confidentiality, authenticity and integrity at link layer. Moreover, the IEEE 802.15.4 standard can be used to protect against message replay attack and can also provide access control mechanisms supported with access control lists (ACL) [9].

At the **application layer**, there are multitude of network protocols optimised for use in local constrained device networks such as Constrained Application Protocol (CoAP) [10], Message Queue Telemetry Transport (MQTT) [11] and Extensible Messaging and Presence Protocol (XMPP) [12]. Most of these "lightweight" messaging protocols lack strong security features but a number of works have initiated the path to implementing security into them. Examples include OAuth [10] and Open ID Connect implementation [14] [15], DTLS [16] over CoAP, Lithe [17], etc.

Nevertheless, there are still a number of open issues regarding IoT security. Considering the constraints of low-energy consumption and low processing capability as well as scalability factors in IoT environments, the use of cryptography into the devices is still a concern. In most of the cases, the encryption key management mechanisms are still not properly addressing the big issue of dynamicity (scalability in and out) and diversity of things. The key management at the perception layer is a critical issue to be solved in order to address security; this includes key generation, distribution, storage, updating and destruction processes. IoT lightweight key management schemes are required such as key broadcast distribution in the entire network, group key distribution and master key distribution [6]. Works such as [18] propose the adaptation of DTLS to enable group keys in multicast communications using CoAP. In [19] the authors state that most of the existing Key Management Systems (KMS) are

not suitable for IoT. The KMS suitable in IoT environments are those that support low computational overhead on the things, in contrast to public key cryptography algorithms. However, there are works based on PKI schemes for IoT [20].

Even if there are communication protocols such as IEEE 802.15.4 that defines the symmetric encryption as the security mechanism, conventional symmetric cryptography such as Advanced Encryption Standard (AES) is not adequate due to: IoT environment constraints [5]; the high complexity of key exchange protocols in scalable environments; and the problem of key confidentiality [6]. On the other hand, lightweight cryptography (LWC) algorithms are promising for IoT [21], for example Elliptic Curve Cryptography (ECC) [22].

Regarding authentication, there are different approaches under research. Zhao [23] proposes a data packet encapsulation mechanism that reduces the overhead of data resources use and combines cross-platform communication features with secure encryption, authentication and signature algorithms to establish a secure communication among things. The work in [24] implements two-way authentication security scheme for IoT based on DTLS protocol and public key cryptography algorithm (specifically based on RSA) designed to be used over UDP/IPv6 over LoWPAN (6LoWPANs) communication stack.

The secure exchange of data also requires the unique identification of the things of the IoT system. As cryptography has been included into IoT in the last years, the cryptography-based identification mechanisms can be used in those cases. But as aforementioned the use of cryptography implies an overhead that sometimes cannot be afforded [25].

Different works identify diverse security and privacy threats of IoT. One of the most prominent is the work of Open Web Application Security Project (OWASP) that identifies the top ten most common vulnerabilities of IoT systems [26] covering the whole IoT architecture layers (from *Insecure Web Interface* to *Poor physical security* flaws). These most common vulnerabilities include insufficient authentication and authorization, insecure network services and lack of transport encryption and integrity verification. Authors in [27] exemplify hands on the "most severe, yet easy to abuse" IoT threats, namely: leakage of the personally identifiable information (PII), leakage of sensitive user information and unauthorised execution of functions. The research done in [28] provides twenty security considerations for cloud-supported IoT, ranging from *Secure communications* to *Impact of Cloud decentralization on security*, and it describes the maturity of the research approaches for addressing them.

From these and similar works [29] [30] [31], it is clear that end-to-end IoT security and privacy are highly challenging. More recently a complete survey on IoT security and privacy challenges was published [32] which comprehensively analyses IoT security challenges of various layers and intrinsic vulnerabilities from the perspective of technologies and architecture used. Further, the lack of IoT architecture standards does not contribute to facilitating security- and privacy-aware design of IoT applications. The work of ENISA in IoT security is currently focused on providing support and guidance for four main domains, namely, airports, cars, homes and cities. Therefore, in the ENACT Elderly care case study special attention will be paid to smart home security best practices [33].

ENISA has also recently produced two valuable guidelines on IoT security. The first one provided the basic recommendations for IoT security in the context of critical infrastructures [34]. The second guideline reports the security issues posed by IoT systems that use Cloud computing technologies and advises a number of security and data protection measures [35].

Considering all of the above, ENACT will research on how to define the needed system and data protection measures in the different layers of the IoT system and how to make sure they work in harmony together for an efficient and holistic situational awareness and security and privacy assurance.

2.2 IoT Security- and privacy-by-design

In the last years, the trend in smart and trustworthy software engineering processes include security- and privacy-by-design practices which prepare the software to be compliant with the needed security and privacy requirements and regulations.

Of outmost importance for EU software industries and system vendors is the compliance with the new European General Data Protection Regulation (GDPR, Regulation (EU) 2016/679) which has been in force since May 2018. GDPR compliance implies both privacy and security mechanisms definition, enforcement and control, including evidence collection for ensuring transparency to end-users, third parties in service provision (if any), and law enforcement authorities.

ENACT is a DevOps framework aimed at tackling security and privacy challenges of design, deployment and operation of IoT systems. To this aim, ENACT proposes a risk-driven analysis (WP2) that will enable the identification of the main privacy and security countermeasures and controls necessary for ensuring system privacy and security behaviour.

2.2.1 IoT Security and Privacy requirements specification

Two major trends can be identified in the literature for the specification of security and privacy related requirements of distributed systems that include the use of Cloud services. A description of the state of the art of each follows in the next two subsections.

2.2.1.1 Model-based specification

This approach consists in expressing the security and privacy requirements on the basis of the architectural model of the IoT system defined at design time. Usually this architectural model describes the distribution of the IoT system components together with their communication and deployment requirements.

According to the security-by-design and privacy-by-design principles, the model can be enriched with the information on required security and privacy behaviour. To this aim, a number of approaches exist for model-based security and privacy requirements specification ranging from annotations over the model elements to more sophisticated security use cases and privacy use cases.

For example, [36] proposes to take advantage of the well-known Model Driven Engineering (MDE) strategy to generate and deploy service-related policies that will be used to take into account non-functional requirements (as security and quality of service) while deploying and monitoring service oriented cloud distributed systems. This allows raising the abstraction level and introducing more automation in software development, improving reusability of: requirements, Platform Independent Models and parts of Platform Specific Models (which give details on the deployment platform). Moreover, MDE is also adapted to define the Model Driven Security (MDS) strategy [37]. MDS defines a framework used to generate security policies out of annotated business process models [38]. This approach requires enriching the traditional “as a service” (XaaS) layer model with a “Business as a Service” level, used to express business-dependent performance and security requirements. This approach modifies and increases the complexity of the standard model.

In the cloud context, one prominent alternative to express security and privacy requirements of the IoT system has been the exploitation of the system architectural model in CloudML (*Cloud Modelling Language*) [39][40]. CloudML is an initiative by SINTEF partner in ENACT and it is currently open source. It provides a domain-specific language that supports the specification of provisioning, deployment and adaptation concerns related to cloud-based systems at design-time and their enactment at run-time.

CloudML is inspired by the OMG Model-Driven architecture approach [41] and supports application deployments to be specified in terms of cloud provider independent models (CPIM) that are later refined into cloud provider-specific models (CPSM) depending on deployment choices.

CloudML was also core part of Cloud Application Modelling and Execution Language (CAMEL) [42] language, a family of domain-specific languages (DSLs) defined in the PaaSage EU project [43] in order to cover the necessary aspects of the modelling and execution of cross-cloud applications.

The CAMEL language [44] integrates and extends existing DSLs, namely Cloud Modelling Language (CloudML) [39][40], Saloon, and the organisation part of CERIF [46]. In addition, CAMEL integrates new DSLs developed within PaaSage, such as the Scalability Rule Language (SRL) [47] and new features (e.g., WS-Agreement parts etc.). In general, CloudML is used to describe the cloud-based application structure and specify the topology of virtual machines and application components [48]. In brief, the key modelling elements that CAMEL shares with CloudML are: Cloud, VM type and VM instance, Internal component, Hosting and Hosting Instance, Communication and Communication Instance.

CAMEL was extended in the MUSA project [49] with deployment and security features required by multi-cloud applications [50]. The extensions were made on the CloudML language integrated within CAMEL.Other languages exist for IoT cloud system model description too such as *Topology and Orchestration Specification for Cloud Applications* (TOSCA) specification [51]. TOSCA is an open standard that provides a language to describe service components, their relationships and topology, similarly to CloudML. In fact, CloudML is listed as one of the TOSCA compliant tools.

In WP2 for the orchestration and deployment of SIS, ENACT will develop a tool called GeneSIS framework (see deliverable D2.1) for the Orchestration and continuous deployment of SIS. The framework will include a modelling language, named GeneSIS, to support the specification of SIS deployment models. This language will inspire from the CloudML language. The main reasons for selecting CloudML over TOSCA for GeneSIS creation are mainly its simplicity and the fact that it has support for natively representing runtime information (not available in TOSCA).

The plans of ENACT include to study the viability of the definition of security and privacy requirements in both GeneSIS and ThingML [52], which is also used by GeneSIS framework to express device level behaviour. These definition in ENACT will take profit of the path initiated by MUSA project which extended the PaaSage version of CloudML in CAMEL language with both multi-cloud deployment and security aspects.

2.2.1.2 Service Level Agreement (SLA)-based specification

This approach relies in the design time specification of privacy and security level objectives, representing the system's Service Level Agreement (SLA) which will be continuously monitored at runtime to ensure that the SLA is metThe standard ISO/IEC 20000-1 [53] defines a Service Level Agreement (SLA) as a *documented agreement between the service provider and customer that identifies services and service level objectives* (SLOs). With the terms *Security SLA* and *Privacy SLA* or *Privacy Level Agreement* (PLA) we therefore respectively refer to the agreements that specify *security level objectives* and *privacy level objectives* offered by a service, which can be considered as part of an overall SLA or as complementary to agreements on other service level objectives, such as quality or performance SLOs.

Therefore, an *SLA* defines the *Service Level Objectives (SLOs)* and associated *controls*. *Controls* ensure that the service's and/or the service provider organisation's *capabilities* satisfy the necessary *requirements* derived from the *policies*, which can range from regulations (like GDPR) to organisational policies or orders. The *SLOs* are expressed in terms of *metrics* to quantitative and unambiguously specify the capability levels guaranteed in the SLA. Therefore, Security SLAs associate to each service both the *security controls* that are implemented on top of it and the *Service Level Objectives (SLOs)* of the *security capabilities* of the service and its provider.

The most complete and detailed standard security control family, the NIST Security and Privacy Control Framework NIST SP-800-53, revision 5 Draft [54], provides a comprehensive collection of security and privacy controls that an organisation and/or service can offer. The revision 5 Draft [54] would be therefore used to define the required controls. This revision extends the previous version of the

framework and defines, in addition to *security controls*, *privacy controls* that are specifically devoted to meet privacy requirements and to manage the privacy risks in an organisation, and *joint controls* that can meet privacy and security requirements. Security controls are defined by NIST as *the safeguards or countermeasures prescribed for an information system or an organization to protect the confidentiality, integrity, and availability of the system and its information*, while privacy controls are *the administrative, technical, and physical safeguards employed within an agency to ensure compliance with applicable privacy requirements and manage privacy risks*.

NIST organises the controls in families, such as Access Control (AC), Identification and Authentication (IA), Risk Assessment (RA), System and Communications Protection (SC), System and Information Integrity (SI), etc. And a new Privacy Authorization (PA) family has been added.

The definition of the required control set should be the result of the risk analysis phase. According to the identified threats against the IoT system and the risk profile of the organisation, threats can be classified as those requiring treatment (high and medium risk level) and those that may not require treatment (low risk level or risk accepted). Then, the DevOps team would indicate the security controls and privacy controls that are the treatments to mitigate the identified threats.

Each of the controls should be associated with a level objective in form of a corresponding set of metrics that quantify the fulfilment of the control that can be guaranteed to the service customers. The last task would be to formally express such controls and corresponding metrics in a machine-readable Security SLA format such as WS-Agreement for easing automatic monitoring at run-time.

For multi-component distributed applications, as most of the SIS are, the creation of the Security SLA of the whole application involves the understanding and considering of the dependencies of the components among them and with the external services they may use. In the context of SIS, these external services used can be Cloud services or services offered by “black box devices” which are not under control of the developers. The ultimate goal is to obtain an SLA that includes the security controls that can be granted by the distributed application to its consumers to be later monitored at run-time. Such Composed SLA in fact is the collection of the set of controls that can be effectively promised for each application component. The controls are security and/or privacy mechanisms implemented by the component or required on the Cloud service used. For multi-cloud based applications, the authors in [55] propose a complete methodology for Security SLA composition developed in MUSA project.

ENACT WP4 will study the needs of the use cases for automation of security and privacy controls definition in a machine-readable format that can ease and focus the security and privacy assessment of the running IoT system. The study will involve the usage of MUSA threats catalogue [56] to specify IoT threats identified by the IoT systems under study as well as the definition of required associated controls.

2.3 IoT Security and Privacy Assurance

Ensuring the confidentiality, integrity, and availability of information being processed, stored and transmitted are high-priority concerns in IoT systems. Assurance of these capabilities involves the monitoring of potential attacks and incidents (threats) in order to make sure that the capabilities hold during IoT system run-time. In case information security events or potential deviations from designed secure and privacy-respectful behaviour are detected in the IoT system, a prompt reaction will be necessary, sometimes involving the re-design, re-configuration or re-deployment of system elements. The DevOps approach adopted in ENACT is expected to enable this in an agile manner.

2.3.1 Monitoring

Different works identify the diverse security and privacy threats of IoT systems. One of the most prominent is the work of Open Web Application Security Project (OWASP) that identifies the top ten most common vulnerabilities of IoT systems [26] covering the whole IoT architecture layers (from

Insecure Web Interface to *Poor physical security* flaws). Authors in [57] exemplify hands on the "most severe, yet easy to abuse" IoT threats, namely: leakage of personally identifiable information (PII), leakage of sensitive user information and unauthorized execution of functions. Cvitic et al. [58] analysed the security aspects for each layer of the IoT architecture: the biggest security risk is at perception layer of the IoT architecture due to the specific limitations of devices and the transmission technology used at this layer, followed by the middleware layer based on cloud computing and inherited vulnerabilities of the concept. Mahmud et al. [59] have stated that several IoT security issues might be unnoticed or poorly addressed by researchers, as this paradigm is not full-fledged. A key requirement identified is access control: to ensure that an authenticated IoT node accesses only what it is authorized to.

In such complex IoT threat landscape developing monitoring mechanisms able to detect security anomalies (intentioned or accidental), privacy flaws and misbehaviour is not a trivial task.

The assessment of security posture of a complex system like a SIS is a data-intensive activity [60] that requires the collection and processing of data from many different, internal and external, sources (e.g., logs, network data capture, events, etc.). Typically, monitoring needs to collect and process data in a large number of different formats, provided by a disparate set of sources using different data access mechanisms.

Monitoring approaches usually adopt *Event-Driven Architectures* (EDA) [61], which promote the exchange of events via a specialised publish/subscribe middleware (e.g., Apache Kafka [62], RabbitMQ [63]). This approach supports high scalability and flexibility. The event push model uses unidirectional, asynchronous, fire-and-forget communication patterns that require well-defined message semantics [64], as opposed to the SOA pull, synchronous model.

The most widely used tools for monitoring malicious activity or security policy violations are Intrusion Detection Systems (IDS) also known as Intrusion Detection and Prevention Systems (IDPS). Many IDS solutions include or work together with a Security Information and Event Management (SIEM) system to centrally collect and visualize detected violations and incidents. Prominent open source examples of such solutions are OSSEC [65], which is a host-based IDS capable of analysing system logs and configuration changes and reporting anomalies, and OSSIM [65], which is an IDS offering log management as well as asset management and discovery with information from dedicated security controls and detection systems.

According to Zbakh, M. et al. [66] that evaluated multiple IDS architectures for cloud-based systems, differentiate two main approaches for monitored data analysis and detection component of the IDS:

- *Pattern-based techniques* or *signature-based techniques*, which consist in identifying threats by comparison with a set of previously defined threat patterns. Even if these techniques are highly accurate, they are limited to known attack detection.
- *Behaviour-based techniques* or *anomaly-based detection*, which consist in identifying anomalies (i.e., abnormal behaviour) by comparison of new behaviour with a preconstructed model of normal behaviour obtained by using machine learning methods. Detected anomalies can range from point anomaly (when a single data event deviates from dataset), contextual anomaly (when the data event deviates from dataset in a known context) and collective anomaly (a collection of similar data events behave anomalously with respect to the rest of the dataset). The behaviour-based techniques require pre-defined criteria to classify normal vs. suspicious behaviour. The major advantage of these techniques is that they target the complex task of unknown threat detection.

Usually, the combination of both approaches would be required in order to ensure an extensive while accurate detection.

Other approaches rely on monitoring the security controls specified in SLAs in form of metrics over security capabilities offered by the system. In this line, two major open source tools can be found as part of the EU-funded research projects:

- *SPECS Monitoring tool* [67]: The SPECS project aims at delivering an open source framework to offer Security-as-a-Service. The solution offers techniques to systematically manage SLAs life-cycle including automatic negotiation, monitoring and enforcement of SLAs between CSPs

and SPECS platform based on security properties of cloud services. Their monitoring solution is able to monitor security parameters specified in the SLA of a cloud-based service expressed in WS-Agreement format.

- *MUSA Security Assurance Platform* [68]: In MUSA the specification and enforcement of security is also based on SLAs: the security properties are specified in the application SLA and the monitoring and enforcement mechanisms are aligned with them. MUSA offers an open source tool that includes a cloud Monitoring tool based on MMT tool by Montimage company and an Enforcement tool based on external enforcement agents by Tecnalia. The Monitoring tool is able to correlate information from probes deployed at network, system and application levels.

Note that controls specified in the SLA can refer to any layer (network, device, edge, cloud, application), and that monitoring of such controls would require the correlation of information from multiple distributed probes of different nature.

The planned support to security and privacy monitoring in ENACT is fully described in Section 6.1.1.

2.3.2 Enforcement

Enforcement or control of security and privacy behaviour of SIS is a challenging objective also. The main idea behind ENACT approach for enforcement is to automate as much as possible of the controls to ensure secure behaviour and privacy respectful data protection at run-time. The enforcement of the security will partially be covered by the countermeasures specified at design-time, provided they are actually deployed together with the system components. In addition to such preventive security controls, reactive controls will also be required as part of the reaction process when deviations or violations are detected.

A multitude of approaches for enforcement of security controls exist, but usually they focus on ensuring specific security capabilities and do not rely on automation or orchestration of multiple controls at a time. Furthermore, they usually lack links with a previous formal and tool-based analysis of risks and well-defined risk control strategies at design-time [69].

With regards to automation solutions, two major open source approaches to cloud security mechanisms orchestration can be extracted from recent literature. First, the SPECS Broker [69] is able to deploy a set of well-defined external security mechanisms on the basis of the operator decisions. Second, the MUSA Enforcement mechanism within the MUSA Security Assurance Platform [68] was built following EDA architecture. The solution was developed by Tecnalia (partner in ENACT) and includes a set of security agents (IdM, access control, high availability) together with a MUSA Enforcement Dashboard for configuring the agents and managing the events sent by and to them.

Other cloud security solutions like the PRISMACLOUD cryptographic tools [71], which include Secure Object Storage in the cloud, Flexible Authentication with Selective Disclosure, and Data Privacy by anonymization techniques, are still under work and they will not be released as tools but as libraries for use as third party services enhancement.

The planned support to security and privacy enforcement or control in ENACT is fully described in Section 6.1.2 to Section 6.1.5.

2.3.3 Access Control

The academic and industrial state-of-the-art on IoT Access Control for consideration within ENACT are focused on providing access control with dynamic and adaptive capabilities, through context-awareness, as well as with risk and trust as potential sources of context information.

Dynamic access control. Mahmud et al. [72] have stated that several IoT-centric security issues might be unnoticed or poorly addressed by the security researchers, as this paradigm is not full-fledged yet. A key requirement they identified is access control: the act of ensuring that an authenticated IoT node accesses only what it is authorized to. Cvitic et al. [73] analysed the security aspects for each layer of the IoT architecture: the biggest security risk is at the perception layer of the IoT architecture due to the specific limitations of devices and the transmission technology used at this layer, followed by the middleware layer based on cloud computing and inherited vulnerabilities of that concept. Fall et al. [74] have learned that cloud computing infrastructures do not use dynamic access control, but static traditional mechanisms, despite the highly dynamic nature of cloud computing capabilities. Farooq et al. [75] confirmed that, in the future, more security techniques (such as risk assessment) must be explored in each architectural layer.

Context awareness. Jagadamba et al. [76] studied adaptive security schemes based on context. Context-awareness enhances the effectiveness of the mechanisms by incorporating contextual data into a decision-making process. This capability of taking grey decisions instead of black-or-white is particularly key in environments where perimeter security is not enough anymore, especially for cloud and IoT infrastructures. Habib et al. [77] have identified 3 types of context (physical, computing, user-related), with 4 approaches (category, context-awareness, context learning, context modelling). Interestingly they identified active or passive context awareness (contextual changes are automatically discovered or statically presented), as well as sensed (taken from the processes' environment) and derived (computed on the go).

Risk-based access control. Dankar et al. [78] learned that different risk classes are identified ahead of time and each class is matched with a protection level. An access request to a resource undergoes automated risk assessment and is classified into one of the predefined classes accordingly. The appropriate protection level is then applied to the requested data.

While analysing competing smart home frameworks, Fernandes et al. [79] refined this by considering that device operations are inherently asymmetric risk-wise and a capability model needs to split such operations into equivalence classes. An on/off operation pair for a light bulb is less risky than the same operation pair for an alarm. They proposed splitting/grouping objects' capabilities based on risk, hence with the possibility to select the granularity. From the range of granularities observed, none was risk-based.

Fall et al. [74] learned that many researchers define a risk formula for a given user or object, but on an insufficient set of parameters (e.g., focusing on requestor but not on the resource accessed). They learned also that the main issue with risk-aware access control is the cost of computation. The benefit is that risk is evaluated for each access request, but this is costly in terms of computation. Their proposition does not solve the issue.

Privacy concerns. Hiller et al. [80] put a focus on involving privacy in risk management, while analysing the NIST Privacy Framework. Privacy is an essential part of planning for cyber secure systems, especially during crisis management, when privacy and personal integrity issues can be overlooked. They expressed privacy risk as the product of the likelihood of a problematic data action which may cause a negative individual impact (for instance appropriation, distortion, induced disclosure, insecurity, surveillance, unanticipated revelation and unwarranted restriction on personal information) and the impact of the problematic data action. They also confirmed that adaptive capability is a cornerstone of resilient systems, and therefore contributes to resilience of privacy.

Contribution of trust. Jagadamba et al. [76] learned that conceptually, *trust* is a parameter, used to exchange information regarding the entities actions through belief and faith. Positive behaviours increase the trust, and negative behaviours decrease the trust upon the entity. Trust is classified into *proofs* (certified information –such as identity, property and authorization- issued by a certification

authority or from other central controlled systems) and *indicators* (possible factors collected from various sources).

The ENACT Context-aware Access Control enabler will deal with these considerations, by providing dynamic access control mechanisms for IoT based on context awareness and risk identification, in order to control access to personal data and then protect privacy. It will contribute to trust by ensuring confidentiality in the data managed by Smart IoT Systems, and by providing security and privacy to Operation phase in order to control the access of all the actors (end-users, services, devices, administrators) to the operated data and resources. See Section 6.1.3 for further details.

2.4 Resilience in IoT systems

The **resilience of a system** refers to its "*capacity to recover quickly from difficulties; toughness.*" [81] For software-based systems, resilience is often defined as "*The ability of an app to recover from certain types of failure and yet remain functional from the customer perspective*." [82]. Resilience is particularly important for IoT systems because such systems are usually built with a large number of inexpensive or even disposable devices, utilizing ad hoc networks, and facing complicated physical environments. Therefore, IoT systems are often exposed to internal failures of both devices and network, as well as external perturbations such as cyber-attacks or physical interference.

Researchers seek the improvement of IoT resilience both at design time and at run-time.

At design time, the system can be more resilient based on a well-designed system architecture, which considers the potential failures and prepare for them with, for example, redundant devices or distributed communications. In his PhD dissertation [83], Kyle Benson reported his experience on improving the resilience level of two large scale IoT projects by architectural design. In these two projects, he identifies the main weak points as the device failures, the unstable network connections, and the incompetence of lightweight communication protocols, in particular, the MQTT protocol, including limited expressiveness and centralized broker architecture. Aiming at these types of failures, they propose an IoT resilience middleware, which exploits 1) the redundancy of devices to achieve functional resilience, 2) the overlay peers, especially with the consideration of geographical properties of peers, to improve network resilience, and 3) the decentralized brokerage architecture to improve the resilience of data exchange.

Designing the architecture of an IoT system with redundant devices is a common design-time strategy to improve both resilience and performance. Similar work can be found in the domain of Wireless Sensor Network (WSN) [84]. Public sensing is also considered as a recent trend to increase the architectural resilience of IoT systems. The main idea is to leverage mobile phones or other general-purpose devices to carry out the sensing tasks, and to utilize the public communication network, such as LTE. The rationale behind the approach is still redundancy (exploiting the large number of potential mobile users as sensors), as well as to outsourcing the network resilience problem to the public network. Pachube [85] is representative approach in this direction. A relevant direction is Participatory Sensing Networks (PSN), which entice the crowds to carry out sensing tasks. The participants may need to use their handy devices, such as mobile phones, but may also use more subjective ways to collect data. The rationale is still to achieve resilience by widening the sensing source. The challenge here is how to motivate the participants and guarantee the correctness of the data. Rewards and reputation systems are important ways to achieve these features, such as the bargain-based mechanisms proposed by Xie [86].

At run-time, the main idea behind the research on IoT resilience is to enable the dynamic adaptability of primarily the devices and the network nodes in the IoT system, following a monitor-analyse-repair model. Oteafy et al. [87] approach of Dynamic Sensor Network targets the resilience in IoT systems by introducing the adaptation capability to both the devices (The Dynamic Core Node) and the network (the Wireless Dynamic Component). The authors formalize the behaviour of both components with a resilience model (which defines what failure to handle and when to handle them) and a reaction model

(how to fix the nodes). Focusing on the resilience of the network behind IoT systems, a recent trend is to utilize virtualization to decouple the hardware from the operational capacities, following the same direction of Software Defined Networks. Wireless network virtualization techniques will be particularly useful in IoT, and a survey of useful techniques in this direction can be found in the journal of IEEE communication surveys. [88]

As a summary, Delic [89] identifies the following techniques as key future directions for IoT resilience, i.e., diversity, adaptation, correlation, causation and renewal. Among them, diversity is mainly effective at design time, whereas the adaptation is mainly used at run-time. Correlation and causation are the mechanisms to analyse and plan what to do in case of failures happen, and renewal is the actual reaction to failures. These mechanisms can be used both at design time and at run-time. Diversity is how the natural systems remain resilient. Recently there are research attentions towards the diversity of software systems. Some representative directions and approaches are briefly introduced in the next section. However, in IoT system in particular, little research effort has been spent in this direction.

2.4.1 Software Diversity

In both nature and society, diversity is a fact that different individuals coexist within a system, such as an ecosystem or an organization. It is considered as a main reason why a system remains resilient [90]. In natural systems, two types of biodiversity are of the most interest to ecologists, i.e., gene diversity, which means that within a species, every single individual is unique, as is coded in its gene, and trophic web (food web) diversity, which means that in a system with species connected via food chains, there are species that can be alternative to each other. The two types of diversities are related to each other. Gene diversity increase the resilience of a species, against environmental changes or diseases. Trophic web diversity makes an ecosystem resilient, even if one or more species extinct or face a significant decrease of its biomasses. The two types of diversity are correlated to each other. On one hand, gene diversity within one species, amplified by the environment, may evolve into multiple alternative species in the ecosystem. On the other hand, if a species is diverse enough to handle environment changes, the ecosystem may be more resilient even without strong trophic web diversity.

Both gene diversity and trophic web diversity are inspiring software researchers, with slightly different focuses.

In the software domain, gene diversity corresponds to functionally identical components with diverse code. N-Version programming, or N-Version design, is a long-term research topic and industry practice in software engineering, with the focus on software security. N-Version design is defined as "the independent generation of $N \geq 2$ functionally equivalent programs from the same initial specification" [91]. This is usually done by different development teams coding separately under the same specification. The main idea of N-Version design remains the same after decades of evolution, but there are still challenges to guarantee the true diversity among the N versions, from the perspective of process, organization and even culture [92]. Recent approaches are also focused on the specific domains, such as the multiple versions of firewalls [93]. One of the main drawback of N-Version design is the cost: It requires N times more resource to implement a feature. Automatic randomization is a research direction to address this issue. Static randomization takes the same source code or model as input and produces multiple diverse programs. Forrest et al. defines the two major methods to randomize the compiling result, i.e., randomly adding or deleting nonfunction code, or reordering code [94]. Such randomization at instruction-set level is implemented by randomly mapping between artificial CPU instructions and the real ones [95]. In some execution environments, the "no operations", such as NOP in X86, can be used to randomize the compiling results [96]. The main purpose of such randomization is to increase security [97].

The trophic web diversity inspires software researchers in utilizing and managing the naturally existing diversity of software components to achieve a diverse software architecture. In software industry, there are different software solutions that provide similar functionalities, such as operating systems and browsers, and more in general the large number of off-the-shelf components. In addition, software components or solutions are often customizable or configuration, resulting in many diverse and alternative software components. Hiltune et al. [98]propose the Cactus mechanism that relies on fine-

grained customization of different components and the adaptation capabilities of these components to achieve survivability, i.e., to tolerant the unpredicted events. Caballero et al. [99] utilizes the existing diverse router technologies to design the network topology with diverse routing infrastructures. Totel et al. [100] exploits the fact the COTS (components off the shelf) for database management and web servers have very few common mode failures [101], and designed the experiments with web servers made by diverse COTS. The results show that the proper exploitation of nature diversity contributes to intrusion-tolerant systems.

In ENACT we will exploit the automatic generated software diversity to improve the resilience of IoT systems. The state of the art of software resilience is focused on the reactive way of renewal of failed node or system, and in most cases, the renewed components or subsystems are identical to the failed ones, which are potentially exposed to the same threats. ENACT will improve this by proactively providing diverse components or subsystems so that failed parts can be renewed into a safer alternative. Software diversity will in general increase the complexity of system development, deployment and monitoring. In ENACT, we meet this challenge by investigating on the automatic generation of diverse components and architecture, and the integration of such automatic generation into the DevOps processes. See Section 5.2 and Section 6.2 for further details.

3 Analysis of Use cases requirements over WP4

The analysis of use cases carried out within WP1 produced a number of usage scenarios from where the requirements of ENACT solution components were derived. Such analysis was complemented in WP4 with a dedicated questionnaire that inquired about different security and privacy aspects that clarified the needs of the use cases in terms of (personal) data protection both at rest and in transfer. In the following sections we summarise the main aspects and threats that will impact the design of WP4 methods and the collection of formal requirements that will be used for evaluating the success of WP4 solutions.

3.1 Security and privacy aspects and threats in use cases

In the following table the security and privacy related elements of each of the use cases in ENACT are summarised.

Table 2. Security and privacy related aspects in ENACT use cases.

Security & Privacy aspect	UC1 - ITS	UC2 - eHealth	UC3 – Smart Building
Communication Protocols	6LowPan, Wifi, RFID, ZigBee, and TCP/IP for GW-Cloud comm.	Bluetooth 4, Serial (over USB), MQTT and REST over IP (Ethernet or WIFI)	Z-Wave in the IoT Smart Space and Modbus TCP in Building Control.
Sensors	RFID tags, accelerometer, RSSI detectors, GNSS receivers.	Medical devices (typically bluetooth). Tracking devices. Smartphones. Video Cameras. Environmental sensors.	Temperature, smoke, flood, energy consumption, etc.
Actuators	LEDS as alarms.	Acoustic alarms.	Fancoil, lightning, alarms, etc.
Personal data	N/A	Gateway Id	Presence
Device Identification mechanism	N/A	Device Id - can be MAC addresses (for bluetooth), IMEI (for mobile devices), or UUIDs	Device Id
User Identification mechanism	User name	User names (typically email) or personal national number. Phone number in some applications.	User name
Device Authentication mechanism	SASL authentication mechanisms and LDAP/SSO	Protection on the network to allow connections only for registered devices. Device APIs only allow post.	Default in Z-Wave and Modbus

User Authentication mechanism	SASL authentication mechanisms and LDAP/SSO	Depending on the criticality of the application: user/passwd, user/passwd + fixed IP, National Autentication provided SAML 2.00 (BankID, MinID, ByPass, …).	User/paswd
Device Access Control mechanism	N/A	Under development.	Configuration of list of devices linked to GW
App Access Control mechanism	N/A	Depending on the criticality of the application: user/paswd, user/paswd + fixed IP, National Autentication provided SAML 2.00 (BankID, MinID, ByPass, …).	User/paswd
Roles for trust	Developer, Deployer, Monitoring Operator, Business.	Yes, fine grained permissions are defined in the platform. Each application typically defines its own roles.	No

With the aim to clarify how ENACT can better support the security and privacy needs of the use cases, the use case providers performed a preliminary analysis of the main relevant security threats over their SIS. To this aim, the well-known OWASP IoT Top 10 [26] risks classification was used for identifying the major threats. OWASP IoT Top 10 is an open project that since 2014 collects the most common risks in IoT systems. OWASP Top 10 2017 for web services and its equivalent for IoT systems have become the *de facto* security standards for a basic analysis of risks.

In the following, we provide the conclusions of the preliminary analysis made on identified threats. Note that these threats may, to some extent vary when the use cases perform the risk assessment in the future following methodology developed by ENACT in WP2. The main reasons for potential variations strive on the fact that the threat classification model used may not be the same and that use cases may have already developed or adopted countermeasures for the threats.

Threats in Use Case 1 – Industrial Transport System:

1. Insecure Web Interface

The current access to the Gateways can be done through a non-secure scenario or a wired infrastructure. The connection is done using SSL besides several authentication certificates.

The specification to add security capabilities to the data report functionality is explained. The connections between the different GWs is done considering different types of scenarios.

- The connection between GWs and sensors is done in an On Track scenario that can be considered unsecure and in an On Board scenario which counts with physical protection. The

connection in both cases is based on MQTT, which can be considered secured as TLS and different authentication methods, based on certifications, are implemented. Therefore, it can be considered that the On Track scenario may be potentially endangered as the environment is not under total control.

- The connection between the GWs and the Cloud is done throw an AMQP connector. The connection is based on an AMQP server using LDAP and a Single Sign-On for the authentication issues (only configurable by the system administrator). The scenario is controlled currently as it is not possible to access to the certification if the GW MAC is not registered and the SSL is not broken. The scenario, equal to the On Track scenario exposed above, is not under control, a security failure could be found in case of a device could be physically attacked and the keys extracted. However, it can be considered that the securization can be covered except for the mentioned threats

Summarizing, the scenarios can be considered secured except for not secure scenarios that may infer in a failure in the system. It is expected, into the ENACT context, that the Security and Privacy monitoring would be able to track possible threats in this context to avoid. These points can be also considered for the point 6 Insecure Mobile Interface.

2. Insufficient Authentication/Authorization

All the scenarios of the Rail Use Case are characterized by the following structure.

- 1. The sensors actuator are connected to the Things coordinator nodes, which are identified with an ID besides its own MAC address (used to filter the devices that are connected to the GWs). The link between the sensors/actuators and the node is done in the Thing deployment; no additional sensors/actuators are joined during the operations. The communications between the sensors/actuation and Thing nodes is dependent of the WSN provider. Hence, the communication restrictions may not be equal to the restrictions of the GWs' communications, including the encryption.

All the Things and GWs have a physical MAC address managed by INDRA. It is also stated an ID for the Gateway. The communications between these two entities is done using the MQTT protocol. Therefore, TLS and authentication certificates can be used to secure these communications, as they are available for MQTT. Note that Ethernet cable communications between the Things and the GWs is also considered using rail standardized ports.

- 2. The Things are communicated with the GWs by MQTT. This protocol is valid for safe environments as it is stated in the On Board scenario. However, the On Track communications may be a potential failure scenario, as not all the conditions are under control. AMQP is not implement in this architecture segment as its requirements exceed the nominal capabilities of the Things. Hence, the encryption could be not cover possible attacks to this segment.

All the GWs have a single MAC address managed by INDRA. The MAC address is used to filter the GWs discovered. The communication between the GWs is encrypted based on WPA2 and PEAP certificates are used. The communication with the Cloud follows the same scheme.

- 3. The AMQP communication of this segment is more robust than the MQTT protocol. Therefore, it can cover different attacks based on a stronger authentication protocols. However, as the link between the GWS/Cloud may not be potentially safe, an intruder could get authentication certifies and encryption keys, generating a failure in the system. It is possible regenerating the certifies. However, the problem could remain.

Summarizing all the infrastructure is protected under authentication and TLS/SSL protocols. However, it is considered that attacks, through authentication capabilities, may happened.

3. Insecure Network Services

Based on the explanation of the point 2, the GWs filter the MAC addresses of the devices, which are connected to the GWs. This means that no intruders can be connected to the GWs as a specific MAC is required and TLS/SSL is needed to receive the authentication certificates. In case of being authenticated, the LDAP certificates can be restarted by a third certification entity.

Due to the public AMQP ports are blocked, a GW cannot be substituted or an unauthorized Cloud request cannot be admitted as the same certificates and SSL/TLS are required. Therefore, the GW would not be affected by the DoS attacks in every interface.

However, the scenarios are not completely secured due to its nature. As it was explained in the points 2 and 3, the security highly depends on the scenarios and the protocols stack implemented on each of them. A DoS attack to a GW can be real if the key and certifications are obtained in a single machine and the links to that GW disabled. Moreover, a Cloud DoS attack may generate a failure in the certification system generating a failure in the entire system due to the authentication failure.

The WP4 tools will be used in any case to track possible failures into the Rail system that it is implemented, to detect the mentioned attacks.

4. Lack of Transport Encryption

As it was explained in the points 2 and 3, different scenarios with different protocols, implemented based on the nature of the scenario and the technology available, are implemented. Hence, different security measurements are available on each of them, including the encryption.

The data encryption can be found in several layers of the system. The physical layer (based on IEEE standards) have several encryption methods.

- All the ZigBee devices support AES 128 bits encryption. Authentication capabilities can be also enabled. It must be added that other several MAC access preventions are implemented.
- All the GWs support Wi-Fi communications implementing WPA2 protection systems that includes a 4-way handshake that manages the keys and AES encryption. It must be added that other several MAC access preventions are implemented.

The data ontology also exploits CRC encryption capabilities to ensure the integrity of the data. All the MQTT headers are pass-through CRC methods and the payload is pass-through a single 32 CRC process for no safety data and two 32 CRC processes in case of safety data as it is indicated in the rail regulation.

The data encryption of each level is not unified; the chain does not ensure the integrity of the data as the encryption is changed in the different stages, which can represent a failure.

The ZigBee protocol can implement encryption or not depending on the configuration that the WSN provider states. This open WSN provider criterion generates a possible failure in the change if the data is affected in the Thing level.

These threats, are identified in the encryption section to be tracked by the tools.

5. Poor Physical Security

The system is spread along a wide are in different locations following the rail mapping distribution. These devices are generally designed to cover all possible scenarios.

Several of the infrastructure is physically secured. These cases are for the centralized On Track systems as the On Track GWs and the On Board systems. However, several On Track edge devices are exposed. The mitigation measurements taken for these cases is the IP68 encapsulation design around these devices. This prevents the physical degradation due to the environment and undesired intrusions. However, in the ENACT project, it is expected from the Security and Privacy Monitoring tool to track

usual traffic patterns or undesired instructions (logs) and the isolation of that part of the system is expected to be done by the Security and Privacy Control in case of non-safe applications.

6. Insecure interfaces.

The IoT applications in the Rail Use Case, be they mobile or web applications, will need to include secure interfaces and HMI.

Threats in Use Case 2 – eHealth:

The eHealth IoT system was designed from the very beginning following security-by-design, privacy-by-design and privacy-by-default principles.

As described in deliverable D1.1 of ENACT, the medical Gateway is in the core of the architecture, controlling the edge and devices in the IoT space and providing the connection to the cloud (Tellu Cloud platform).

1. Insecure web services and insecure network services.

Not applicable. The web services used by the Gateway (Raspberry Pi) to Access the information of the sensors and actuators in the environment applies encrypted BLE.

The gateway is in charge of collecting the measurements from the devices and transmitting them to the backend. The gateway is designed to be powered and connected to the backend all the time. It is listening for measurements coming from the devices it is paired with. The pairing process is made prior to system deployment. The gateway is not scanning for other Bluetooth devices and will not accept connections from other devices. It is connected to the internet using an internal 4G modem.

If the patient home is not covered with a compatible 4G signal, the gateway can be connected via WIFI to the home network. In any cases, the gateway itself is behind a NAT and firewall to ensure that it is not accepting any incoming connection from the internet. It communicates to the backend (ActiveMQ) and connects to an administration VPN to be remotely administered. Over the VPN, the gateways expose an SSH server (on port 22) which allows to login using an administration private key. Logging in with user and password is disabled.

2. Lack of transport encryption.

Not applicable. All communications in the system are using the encryption modes in the protocols. Therefore, confidentiality and integrity in communications is ensured.

The gateway uses BLE to collect measurement from the medical devices. During the paring of the device with the gateway (which is done before the gateway is given to a patient), the device and gateway generate and exchange an encryption key. This key is used to encrypt all the communications from the gateway and devices in order to protect the data which is exchanged. Moreover, all communication in-between the main nodes are encrypted using SSL.

3. Insufficient Authentication of devices.

Not applicable. Each gateway is authenticated with a username and password which gives it permission to post data only in its specific topic, i.e. one gateway cannot post in another gateway topic and cannot retrieve and event from other gateways. This is important to make sure that even if one gateway is compromised, its credentials do not give access to any other parts of the system. The gateway user name is the gateway host name and each password is a randomly generated password. The Edge is authenticated with a username and password and it has access to data coming from all gateways.

4. Insufficient Authentication of users.

Not applicable. The authentication is done through national level 4 Authentication server (2 factor authentication with BankID) in order to guarantee high level of security. Once logged in, the user app uses a token over HTTPS to access the TelluCloud APIs. Each nurse is only given access to its own set of patients (through the APIs).

5. Privacy Concerns

Each nurse is only given access to its own set of patients. Patients permissions are setup so that they can only access their own data over the API. Once the data are retrieved and pushed forward by the gateway the measurements are deleted from the device. The system architecture and software are developed in accordance to the GDPR and includes state of the art security features to ensure the protection of the patient personal and medical data.

6. Insecure interfaces.

The IoT applications in eHealth, be they mobile or web applications, will need to include secure interfaces and HMI.

7. Denial of Service.

The Gateway and the TelluCloud platform may suffer DoS attacks.

Threats in Use Case 3 – Smart Building:

1. Insecure web services.

The web services used by the Gateway (Raspberry Pi) to Access the information of the sensors and actuators in the environment (Z-Wave mostly) are using REST API that needs to be secured. Similarly, for the PLC in the Smart Building, the services need to be secured.

Currently, the Gateway is accessible from such REST API and the security is only implemented in the local network and through the Internet proxy that prevents accessing the Gateway.

Due to the fact that in ENACT all communications in the Smart building between the Gateway and the devices will go through the SMOOL IoT Platform, it would be possible to deactivate such web services except for the internal processes that run inside the Gateway.

2. Lack of transport encryption.

Currently, none of the communications uses encrypted protocols. The wireless Z-Wave communication between the Gateway and the devices is not encrypted and in general, even if Z-Wave protocol supports encryption, this mode is not used.

The main reasons for not using encryption in the Gateway communication configuration are:

- Not all the Z-Wave devices (sensors and actuators) support encryption, though some of them support it and could be exploited.
- The limited range of Z-Wave signals would imply a physical access to the network to be able to attack it. Z-Wave devices can communicate point-to-point up to a distance of about 30 meters. Nevertheless, some sensors and actuators can act as signal re-transmitters with ability to hop signals, so effective ranges of up to 180 meters are easily achieved.
- The Z-Wave devices are usually used in smart building and home automation applications and normally the lights or thermostat at home are not an attractive target for attackers.

The communications between the PLC with the sensors and actuators are all cable communications that use communication buses: KNX, Modbus, DALI, etc. Some of these protocols do have their own security mechanisms, for example in a KNX network it is necessary to be programmed individually the devices so as to connect one another. In any case, in order to be able to read the transmission frames that travel through the communication bus it would be necessary a physical connection by cable with the bus.

The communication between the Gateway and PLC with the IoT applications is made through SMOOL and TCP/IP cable connection.

Initially, it is not envisaged the need of encrypting communications. However, an analysis of which are the most sensitive Z-Wave devices in terms of risks when their messages are eavesdropped or intercepted could be made. The communications for the most sensitive devices could be switched to encrypted mode. This way message injection attacks (e.g., attempts to modify the sensed value or the actuation order) could be prevented.

3. Insufficient Authentication of devices.

Each Z-Wave network is identified by a Network ID or Home ID and each device is further identified by a Node ID. Every time a Z-Wave device joins a Z-Wave network a Node ID is assigned to it. There is always a master device or "Primary controller". In the Smart building use case the Gateway (Raspberry Pi) and the "slave" devices would be the sensors and actuators.

Nodes with different Home IDs cannot communicate with each other, but they may have a similar Node ID. This is because the two networks are isolated from each other.

On a single network (one Home ID) two nodes cannot have identical Node IDs. This means each node can be individually addressed.

In general, a pairing or endorsement process between the Gateway and the devices is required just after the deployment. During the pairing process the Gateway is open to accept any device that wants to join the Z-Wave network and it assigns an authorised Node ID to it, which can be considered as a security limitation. The operator in charge of configuring the network through a physical button in the collector activates the paring process. The duration of the process needs to be limited to the minimum so as unwanted devices do not have the possibility to join the network. The limit of 30 meters of distance to the collector is also an additional security measure. After this, the Gateway reads all input messages but it only processes those coming from devices with authorised Node IDs.

4. Insufficient Authentication of users.

Currently there is no interface for humans in the KUBIK Smart Building management system so there is no user authentication made. The IoT applications used in the Smart Building for building management and user comfort would only authenticate the users by user name and password (1 factor only).

5. Insecure interfaces.

The IoT applications in the Smart Building, be they mobile or web applications, will need to include secure interfaces and HMI.

6. Denial of Service.

The web services of the Gateway and the PLC may suffer DoS attacks. SMOOL server could also be target of DoS.

3.2 Requirements to security, privacy and resilience tools in ENACT

Together with the initial analysis of security and privacy aspects, the use cases identified a number of requirements related to the Trustworthiness support in ENACT. The next table summarizes such ENACT requirements that are relevant for WP4 methods and tools. In particular, the considered requirements refer to the following aspects: (i) context aware access control, (ii) software diversity of IoT systems, and (iii) privacy and security monitoring and control.

As it can be seen, in the eyes of the end-users in ENACT, all requirements are high or medium priority and all medium priority ones are recommended or nice to have features. These features will be addressed during the project lifetime though the focus of WP4 work will be on addressing high and mandatory requirements.

Table 3. Requirements for the IoT Trustworthiness support in ENACT

ID	Statement	Source[2]	Brief description	Priority[3]	Need[4]	How to Address in WP4/IoT app
DO-4	***4. ENACT Trustworthiness Toolkit***					
DO-4.1	***4.1. Robustness & Resilience Enabler***					
DO-4.1.1	Gateway recovery and factory reset	2	There is a need to allow for resetting the Medical Gateway to factory default when something goes wrong, and then get the GW operational after reset	H	M	Diversifier
DO-4.1.2	Handle Medical Gateway failure situations	2	The Medical Gateway should quickly recover from being unresponsive. In addition to extensive and continuous testing this, includes features for handling the failure for example through remote access in a safe mode.	M	R	Diversifier
DO-4.1.3	Roll back configuration	2	In case a deployment of a new configuration fails. The GW should be able to roll back to the previous configuration and notify the Operator	H	M	Diversifier
DO-4.2.x	***4.2. Risk-Driven Decision Support Enabler***					Addressed by WP2.

[2] Source - 1: ITS use case, 2: Digital Health use case; 3: Smart Building use case.

[3] Priority - H: High; M: Medium, L: Low.

[4] Need - M: Mandatory; R: Recommendation.

DO-4.3	***4.3. Security and Privacy Monitoring and Control Enabler***					
DO-4.3.1	Authentication	1	Authentication procedures are applied to treat every data packet.	H	M	IoT app[5]
DO-4.3.2	Authentication invalid	1	A procedure to deal with invalid authentication of the elements and users must be designed.	H	M	IoT app
DO-4.3.3	Security not variable	1	The security measurements are not adapted if the system is running	H	M	S&P[6] Monitoring Enabler
DO-4.3.4	Authentication levels	1	Several authentication levels would be designed.	M	R	IoT app
DO-4.3.5	Attacks historical	1	An historical of that would be created.	M	R	S&P Monitoring Enabler
DO-4.3.6	Things and On Board GWs identification management	1	The Ids of the system elements must be checked.	H	M	IoT app
DO-4.3.7	Access security	1	The users must be authorised to access to the tool which manage the SW updates.	H	M	IoT app
DO-4.3.8	Orchestration Interface	1	The Monitoring enabler awares the Orchestration of alerts related with a shift in some of the elements performance after processing the data gathered.	H	M	S&P Monitoring Enabler
DO-4.3.10	Alarm thresholds configuration	3	The Trustworthiness Monitoring enabler should enable the user to set the desired thresholds to raise cybersecurity alarms.	H	M	S&P Monitoring Enabler

[5] IoT app – IoT application (particular to the use case).

[6] S&P – Security & Privacy.

DO-4.3.11	Security enforcement	3	The Trustworthiness Monitoring enabler should work together with Trustworthiness Adaptation Enabler which helps reacting to attacks or incidents	H	M	S&P Control Enabler
DO-4.3.12	Protection of person sensitive data	2	Secure data management across IoT edge and cloud is severe as the system typically handle person sensitive data.	H	M	CAAC[7], S&P Control Enabler
DO-4.3.13	Monitoring and control	2	there is a need to do Real-time monitoring of a set of Medical Gateways and to receive proper notifications with useful information in case of errors.	H	M	S&P Monitoring Enabler
DO-4.3.14	Access control	2	Different users and roles should have different level of access. Need support for role based and user based access control. It would also be interesting to look at context aware authorisation (e.g., in an emergency the access may be different than in normal operation	H	M	CAAC
DO-4.3.15	Authentication	2	Various kinds and levels of authentication need to be supported both at the edge side and cloud side. Support for two factor authentication (or similar level) is mandatory for a set of scenarios in the digital health domain	H	M	IoT app
DO-4.3.16	Secure data transmission	2	Confidentiality, integrity, and authentication across IoT, edge and cloud is needed.	H	M	Secure protocols, S&P Monitoring Enabler

[7] CAAC – Context Aware Access Control.

DO-4.3.17	Communication need to be trustworthy in the sense of reliability, availability, integrity and privacy	2	The trustworthiness aspects of communication within digital health is significant for example, in order to not miss any notifications or alarms, you should be always connected to support emergency situations when they occur, the integrity of data is severe and privacy need to be ensured as there is typically person sensitive data involved	H	M	Secure protocols, S&P Monitoring Enabler
DO-4.3.18	Monitoring, Diagnose information and failure detection	2	The system should continuously monitor system performance, suspicious behavior and failures. The monitored data should be analysed to provide informative and understandable diagnosis	M	R	S&P Monitoring Enabler
DO-4.3.19	Full end-to-end security	2	Support for security across the IoT, edge and cloud space from the medical device, through the gateway and all the way to the target stakeholders (e.g., hospitals, Electronic patient journal etc.) is needed	M	R	Secure protocols, S&P Monitoring Enabler, S&P Control Enabler, CAAC.

In the last column of the table we have included a reference to how it is intended to address the requirement fulfilment in ENACT. For some requirements the software components or modules that will be developed in WP4 will address the issue (Diversifier, Security and privacy Monitoring Enabler, Context Aware Access Control Enabler, etc.). See section 4 to understand how each of the modules fits into ENACT Enablers.

In some cases, the requirement is very particular to the particular use case application and ENACT will not address it by an external enabler, thus, it will be up to the IoT application itself to resolve it. For example, ENACT focus is mainly on authorisation and the solution will not include identification and authentication mechanisms as we rely on the IoT application using a dedicated solution or any COTS or as a Service solution from the plethora of Identity and Access Management (IAM) solutions available in the market.

The coverage of the WP1 requirements above will be tracked in WP4 to learn on the success of WP4 development. The future D4.2 and D4.3 will report on the status of the requirements coverage in initial and final implementation of the mechanisms described in the present D4.1.

4 IoT Security, privacy and resilience support in ENACT

This section describes the planned work in ENACT to provide support to security, privacy and resilience aspects of SIS. The support includes mechanisms and tools at both development and operations phases of the DevOps cycle. While Section 5 and Section 6 detail the solutions that will be offered in Development and Operation phases respectively, the present section provides an overview of the mechanisms in the context of the overall ENACT solution and describes how they relate to each other and to other ENACT components.The remainder of this section is structured as follows. First, Section 4.1 presents the overall architecture for security, privacy and resilience support in ENACT. Second, Section 4.2 introduces the security and privacy mechanisms in ENACT. Finally, Section 4.3 describes the diversity of IoT systems as resilience mechanism in ENACT.

4.1 ENACT architecture for IoT Security, privacy and resilience

Trustworthiness in ENACT includes security, privacy and resilience aspects. Resilience is particularly addressed by software diversity of the different elements of the IoT system.

In Figure 3 it is described how WP4 is planning to support these aspects in SIS. As it can be seen, WP4 will produce mechanisms and tools for both Development and Operations phases of the DevOps cycle.

At SIS **development phase**, the focus will be on design step, where security, privacy and resilience requirements will be analysed and specified together with other requirements of the SIS. The idea is that these three aspects **are not an afterthought but considered from the very beginning of the development process**. To this end, the development will include two major steps:

- *Security, privacy and diversity (resilience) requirements specification*. In ENACT we have opted for defining *security, privacy and diversity (resilience) requirements* at the level of system model. When elaborating the architectural model of the SIS describing its components and relationships, i.e., the SIS model, the security and privacy experts and analysts would need to intervene in the process as part of the development team and collaborate in the definition of required security measures at different layers of the system, data protection and access control mechanisms, data anonymization mechanisms (if any required), etc. The SIS model in ENACT is envisaged to be described in GeneSIS language as explained before.
- *Security and privacy controls specification*. A risk assessment process will be carried out (WP2) to derive the major risks of the system and specify the risk profile including the security and privacy countermeasures (controls) necessary in the system to minimise the risks. The controls will need to be selected by the development team on the basis of risk minimisation by matchmaking of security and privacy requirements with available controls at the different layers of the SIS (network, device, edge, cloud, application). As this is a non-trivial process, ENACT intends to simplify it as much as possible and rely on existing and well-known security knowledge (threats, vulnerabilities, controls) catalogues such as that of MUSA (for cloud security threats and controls), NIST (security and privacy controls), CSA (cloud controls), etc.

At SIS **operation phase**, the focus will be to ensure that SIS security and privacy requirements are met by continuous monitoring of defined security and privacy controls. This way, prompt reaction to detected incidents will be possible and risks over the SIS will be under control. Operational support includes:

- *Security and privacy controls monitoring*. The monitoring and analytics tool within the Security and Privacy Monitoring and Control Enabler will be in charge of detecting any potential incident and attack and raise notifications to the IoT applications.
- *Reaction or adaptation to detected incidents*. Three major tools will be delivered:

- Robustness and Resilience Enabler or Diversifier tool. In charge of collaborating with ENACT Orchestration and deployment engine to make sure the different software variants in the SIS elements are deployed when needed.
- The Context-aware Access Control (CAAC) tool will also be part of the Security and Privacy Monitoring and Control Enabler and will be responsible for IoT tailored and context based authorisation mechanism as explained below.
- The Security and Privacy adapter tool within the Security and Privacy Monitoring and Control Enabler will be responsible for activating security and privacy controls in different elements of the SIS (IoT platform, CAAC, configuration of devices, etc.).

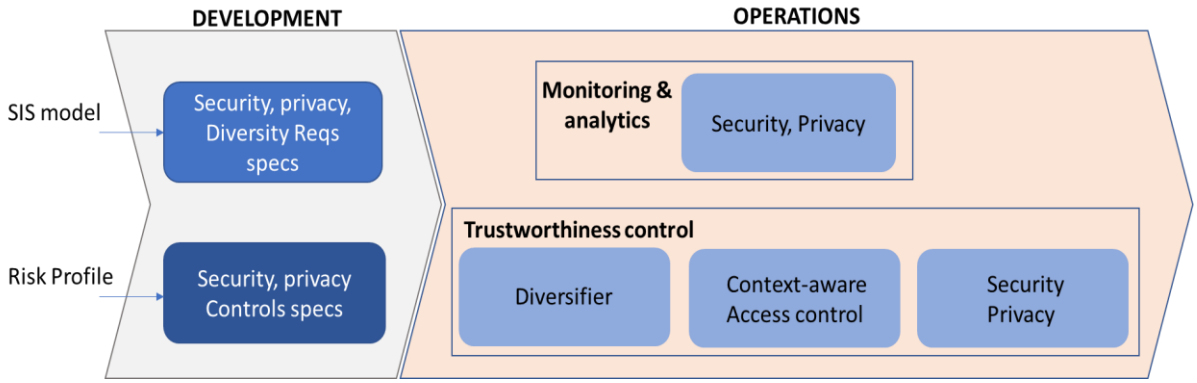

Figure 3 – Security, Privacy and Resilience support in ENACT

In the following Figure 4 we depict the planned WP4 tools (in blue) and their relationship with other operational tools in the ENACT solution.

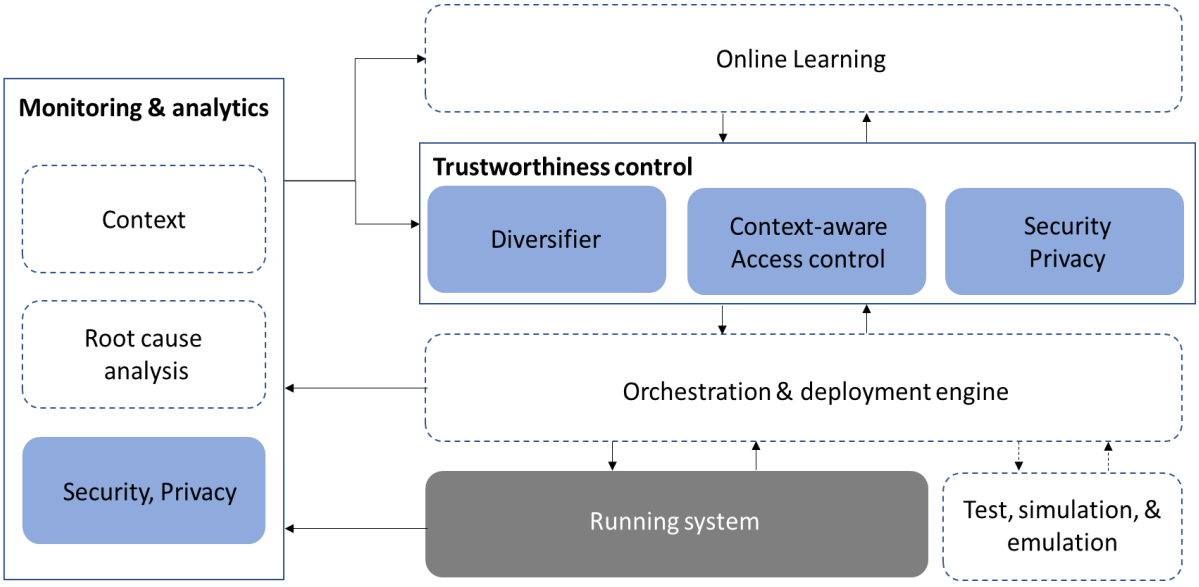

Figure 4 – ENACT Security, Privacy and Resilience tools in Operation

As shown in Figure 2, WP4 trustworthiness adapters or controls will interface with both the Online learning enabler (see deliverable D3.1 for more information on this enabler) and the orchestration engine within the GeneSIS framework (see deliverable D2.1 for more information on this enabler). The WP4

controls will interface with the Online learning enabler for improving security and privacy control efficiency, whereas, they will interface with the orchestration engine for deployment of both software variants and security mechanisms (when they are required).

The Security and Privacy monitoring tool by WP4 will continuously oversee the running SIS and detect any misbehaviour and flaw related to privacy and security of the system elements and communications. It is envisaged that the tool will work quite independently from other performance and context monitoring tools of ENACT because it will offer its own visualisation, notification and analytics modules.

4.2 IoT Security and Privacy mechanisms in ENACT

This section introduces the security and privacy mechanisms initially identified to be offered by ENACT solution. While the first two mechanisms are part of the IoT network and application layers, the last three mechanisms will be part of the Security and Privacy Monitoring and Control Enabler, and are described in Section 6.1.

4.2.1 IoT Communications Security

As it was explained in the use case analysis section, the use cases in ENACT are using diverse wireless protocols for communication with things. In some cases, such protocols do have encryption modes that currently are being used already. In some other cases, even if the encryption mode exists it is not used by the use case. We recommend the use of encryption whenever performance requirements allow it.

At upper layers, HTTPS protocol will be used always.

The support of ENACT for communication encryption will be at the level of detection of encryption mode only. Warning notification will be issued when encryption is not used..

4.2.2 IoT IdM and authentication

Even if identification and authentication of both users and nodes (things, edge, etc.) are fundamental for trustworthy IoT systems, the focus of ENACT project will be on authorisation. ENACT use cases will rely on existing Identification Management (IdM) modules and Identity providers. Thus, authorisation support of ENACT will offer an innovative access control mechanism which includes OAuth 2.0 based authentication.

4.2.3 IoT Context-aware Access Control

The objective of the Context-aware Access Control (CAAC) is to provide mechanisms for controlling the security, privacy and trustworthiness behaviour of smart IoT systems. A specific emphasis will be made on confidentiality and integrity of data and services. This includes reaction models and mechanisms that address the adaptation and recovery of the IoT application operation on the basis of the application context, in order to deliver dynamic authorization based on context for both IT and OT (operational technologies) domains.

The Context-aware Access Control will provide Context-aware risk & trust-based dynamic authorization mechanisms, through an IAM gateway for IoT that includes next-generation authorization mechanisms.

The aim is to ensure that an authenticated IoT node accesses only what it is authorized to.

Access authorizations will be adapted according to contextual information. *Context* may be for instance the date and time an access authorization is requested, or the geolocation of this request; it may be also composed of a set of information about the status of the underlying infrastructure, the physical system status, SIEM alerts, for example to make certain information more widely available in the case when an alarm has been triggered.

By assessing the applicability of OAuth 2.0, the Context-aware Access Control will leverage it as a key protocol for interoperability. Research will address problems of adding dynamicity to the authorization decisions it produces even if OAuth 2.0 is not meant for that, while still a cornerstone scheme for access control. This dynamic capability will be in charge of evaluating contextual information and insert them in authorization decisions.

4.2.4 IoT Platform Security

In IoT environments, one of the most interesting approaches to ensure secure behaviour of the system is to embed security features such as access control, encryption capabilities, etc. into the IoT Platform that captures the sensors data and acts as a gateway to actuators.

As SOFIA is the IoT platform used in two of the ENACT use cases, the approach would mean to have built-in features that enable the platform implement some of the required security and privacy controls.

According to the project use case implementation (see deliverable D5.1 of ENACT), two different versions of SOFIA will be used in ENACT: SOFIA 2.0 (owned by INDRA partner of ENACT) for ITS use case led by INDRA and SMOOL (open source) for Smart Building led by TECNALIA. Both are semantic middleware platforms originated from the same EU funded-research project SOFIA. Both platforms are implemented under the publish/subscribe model that seeks interoperability of heterogeneous devices through the definition of:

- an open API and middleware services based on existing standards that provides a communication back-bone for smart applications,
- a common and extensible data model for smart spaces that enables interoperability among vendors at application (semantic) level and
- a set of design and development support tools that drastically reduces the development time of smart value-added applications.

As described in [102], similarly to SMOOL, SOFIA relies on two main components:

- Knowledge Processors (KP) that are the end points of the smart applications. These components implement the logic of the applications and produce/consume data to fulfil their tasks.
- Semantic Information Broker (SIB) that enables sharing ontology-based semantic information between KPs and acts as gateway that controls whether a message should be transmitted by the TCP/IP stack, Bluetooth, or any other communication technology/network.

Thus adding security features to SOFIA would mean adaptation of these components to make it possible to consider security aspects in the communication between KPs.

4.2.5 IoT Security and Privacy Assurance

IoT Security and Privacy **assurance** refers to the assessment of secure and GDPR compliant behaviour of the SIS. Such assessment involves the **monitoring** of the security and privacy behaviour and detection of deviations as well as the reaction to the deviation by means of **enforcement** of some security and privacy mechanisms that make the SIS recover the secure or privacy respectful behaviour.

ENACT will provide both monitoring and enforcement (control) functionalities in the Security and Privacy Monitoring and Control Enabler. The Enabler is explained in Section 6 and involves a number of tools that all together will support the continuous assurance of the security and privacy requirements expressed at the SIS design.

Thus, the enforcement would include an activation of or a recommendation to use some of the mechanisms above, depending on the case. The enforcement will be automated whenever possible, though this is not always achievable due to the nature of the privacy or security mechanism, which are

usually very interleaved with the application, communication or device at stake. It is not an aim of ENACT to develop novel anonymization techniques. Therefore, whenever the IoT applications do need such obfuscation and anonymization mechanisms we are recommending the use of existing and preferably open source mechanisms such as the ones offered by PRISMACLOUD [68], ESCUDO-CLOUD [103] and CLARUS [104] projects.

4.3 IoT Diversity mechanisms

ENACT aims at improving the resilience of smart IoT systems by promoting the software diversity of these systems. We look to IoT software diversity from two different perspectives, i.e., component diversity and architecture diversity.

4.3.1 Component diversity of IoT systems

Component diversity indicates how component instances differ from each other. It resembles the concept of gene diversity in biological systems. In a bio ecosystem, almost no two individuals are exactly the same in the gene level, even if they are from the same species. Such gene diversity ensures the resilience of species, since a particular external perturbation, such as an environment change and an infectious disease, is difficult to kill all the individuals and thus extinguish the species. In the IoT ecosystem, for the sake of simplicity and maintainability, it is a common practice that the same component is deployed many times in different systems, which causes a large number of identical component instances running in a big ecosystem. This will in turn magnify the effect of perturbations, such as external attack, unexpected user load, or the exposure of software defects.

To improve the resilience of IoT systems, ENACT will investigate the automatic injection of diversity into IoT software components. From the same behaviour specification (the ThingML model) of a software component, the ENACT component diversifier will generate different versions of the component implementation, which are alternative to each other from the behaviour point of view.

As the first step, the ENACT component diversifier will focus on the communication part of IoT components. The difference of the generated component implementations are the communication protocols. In particular, the generated versions provide the same data to the external world, but via different APIs, including the different orders of parameters, the insertion of additional (unused parameters), or the different types of particular parameters. The objective of such diversification is to prevent the potential adversary from cracking all the component instances by learning the behaviour of one instance.

Component diversity will generally increase the complexity of software development, deployment and monitoring. In ENACT, we tackle such complexity by completely relying on the automatic generation and deployment of variant versions of the components. In particular, all the variant components are generated automatically, and developers only need to configure the variation points. Furthermore, the generated components are deployed by the same engine as the original components.

4.3.2 Architecture diversity of IoT systems

The architecture diversity indicates how a local IoT system is different from the other functionally identical systems. A local IoT systems means a functionally self-contained system, composed by devices, gateways and the software components running on them, and is usually used for a single customer, such as a person, a family or a company. A commercial IoT vendor usually provides an IoT solution, which is deployed as multiple copies to their customers. In such cases, each of these copies is a local IoT system, and all these local IoT systems, both the deployed ones and the potential ones, form a big IoT ecosystem provided by the vendor.

The architecture diversity resembles the diversity of a trophic web in bio ecosystems. A trophic web is natural system composed by a number of species connected by energy transferring. One species in the trophic web is fed by some species, and on the same time provide energy to some other species. In a

resilient trophic web, each niche is occupied by multiple alternative species. In other words, each species can live on several other species, and also can be eaten by multiple species. With such diversity, or complexity, if the environmental changes affect the functionally of one species (which means that the biomass of the species are scientifically reduced), the whole system can dynamically adapt to a new balance, with some relevant species changing their food structure by consuming more alternative species.

An IoT system is also an ecosystem, composed by hardware, such as devices and gateways, and software, including platforms, libraries and applications. Each functional niche in the ecosystem, such as temperature measurement, data storage, etc., can be potentially occupied by alternative components. Such component diversity either may come autonomously from the market or be generated based on diversity injection. The IoT system selects a component for each niche, and these selections are usually called a configuration of the system. A resilient IoT system should support a wider configuration space, and is able to switch from one configuration to another at run-time when a perturbation breaks the functionally of an in-use component. However, for the sake of simplicity and maintainability, an IoT vendor often choose to support a very limited configuration space, sometimes only one "default" configuration for all their customers.

ENACT improves the architecture diversity of IoT systems by enlarging the configuration space of IoT systems. At development time, the ENACT architecture diversifier automatically generate new configurations of the IoT system, by attempting alternative components. The generated configurations will be included into the continuously integration pipeline for thorough testing, in order to validate the functionality and the quality of the configurations. At run-time, the ENACT architecture diversifier will provide the generated configurations as the input to the adaptation engine, so that the latter can adapt the system into an alternative configuration when the current one is not working as expected.

The ENACT architecture diversifier will focus on the generation of configuration files that are specified in particular format. As the first step, the diversifier will start from two configuration formats, i.e., Docker and Ansible. Docker allows the deployment of the entire software stack into a virtual image, so that instantiate the identical stack in different resources, either on cloud resources or on local devices. Ansible allows the fine-grained command execution to install and configure such software stacks, either in a container on directly on the operation systems of the target devices. Both configuration formats support the composition of existing pieces. ENACT diversifier will utilize the existing and well tested pieces to generate the ad hoc compositions.

5 Design support to IoT system Security, Privacy and Resilience

This section describes the methods and mechanisms that will be offered by ENACT as security-by-design, privacy-by-design and resilience-by-design techniques to be adopted in IoT system development.

The ENACT support to security and privacy at design time is focused on: i) mechanisms for the specification of both the requirements of the IoT system components with respect to these two aspects, and ii) mechanisms for the specification of the necessary controls (external or internal to the IoT system) that ensure the requirements are met. The approaches proposed for privacy requirements and controls specification are the same as those of security requirements and controls specification, which easies the engineering of the IoT system due to both aspects are addressed similarly in the DevOps process. Therefore, the Section 5.1 provides the description of the mechanisms proposed for security-by-design and privacy-by-design together. Finally, Section 5.2 describes the diversity mechanisms that will be developed for resilience-by-design techniques.

5.1 IoT Security-by-design and Privacy-by-design mechanisms

In order to improve the coherence and efficiency of data protection preventive and reactive measures in IoT systems, security and privacy aspects of the IoT system need to be addressed from the very beginning and not left as an afterthought. In the following we describe how ENACT intends to support developers in the task of specifying at design time the security and privacy requirements of the IoT system under construction (Section 5.1.1) and the security and privacy controls that need to be included in the system (Section 5.1.2) to ensure such requirements are actually fulfilled.

5.1.1 IoT Security and Privacy requirements specification

The initial mechanism of ENACT for security-by-design and privacy-by-design is that security requirements of the system (such as authentication requirements, access control mechanism to use by different elements, encryption mode to use in the communication protocol between system elements, etc.) will be defined at the system architecture model on top of the component relationships and deployment model.

As explained in the state of the art section, ENACT support to security and privacy requirements specification will adopt MUSA project extensions to CloudML[50], which will be further enhanced and tailored to fit IoT needs. This way, together with the performance and functional requirements of the system, the GeneSIS model would be able to express security features required and offered by the system in form of controls.

5.1.2 IoT Security and Privacy controls specification

The specification of security and privacy controls to use by the IoT system shall be the result of a previous risk assessment process which identifies the threats over the system and the desired countermeasures of treatments to minimise the risks. Such treatments at technology level are the controls to include in the SIS, which should be specified as earlier as possible, as often they are software mechanisms that need to be deployed or configured at deployment time.

Therefore, ENACT will try to build on top of MUSA risk assessment methodology [105] in order to derive the needed controls which would need to be expressed in application design models. With respect to controls specification in the architectural model, ENACT will adopt and tailor to SIS the MUSA extensions to CloudML language [50]. The MUSA innovations to CloudML (within CAMEL language) included the enhanced component security behaviour characterization, so as to address concepts required to support both composition of components' security Service Level Agreements (SLAs) and risk analysis. More concretely, they are the following:

- ***Classification of components by their nature.*** This allows to describe what type of service the component is offering (Web, Storage, IDM or Firewall) and how the service is integrated into the overall application (internal component, COST or external security agent).
- ***Security Controls information that properly supports Security Control Framework families***. This allows to specify which security capabilities are required and provided by each multi-cloud application component. The security capabilities are defined in the model by selecting and grouping the security controls part of the capability.

5.2 IoT Diversity-by-design mechanisms

The ENACT diversifier uses one system as input and produces multiple variants of the system. At design time, developers control the diversifier by defining the entire space for diversification and the expected diversity they want to achieve. The former determines in theory how many variants exist for the current system and the latter indicates what and how many variants the developers expect to obtain.

Following this principle, the input of the architecture diversifier at design time are the following:

- The specification of the current system architecture. The specification should be executable, which means we can obtain a runnable system by deploying and configuring software and resources according to the specification. In the first step, the diversifier will target at supporting the ENACT deployment and orchestration language, i.e., the GeneSIS model, as well as one or two mainstream deployment formats, such as Docker specification and Ansible.
- An abstract variability model, with the following contents: 1) a definition about the types of components in the system together with the relationship of these types; 2) a set of fixed component instances and the fixed relations among them; 3) additional constraints on what component instances are allowed as well as the relations between them; and 4) a repository of alternative types of components.
- A quote N about how many variants are expected.

From these inputs, the diversifier will automatically generate N different specifications, in the same format as the first input. The output specifications are also executable, which means that using the same engine, we can automatically obtain N different runnable systems.

The component diversifier with a focus on the communication will take as input the:

- A specification of the communication protocol. In ENACT, we will focus on the support of ThingML as the language for protocol modelling. The model will specify the state transition following the communication events, and the parameters used by the events.
- A number N about how many variants are expected.

The output of the communication diversifier is a set of N different protocol models with different event definition in terms of parameters. From these models, we can use ThingML compiler to generate N different implementations of the protocols which are identical to each other from the behaviour point of view. It is worth noting that for the component diversifier, we don't need a sample implementation as input, as such an implementation can be automatically generated from the sample ThingML model.

Diversity metrics will be provided to quantify how the generated artefacts are different from each other, so that developers can have an intuitive view of diversity generation. For architecture diversity, a potential metrics is the Shannon Index that is widely used in Information Theory and Ecology to measure the system diversity. Shannon Index reveals both the number of different types of components (species) appeared in the system, and the balance of distribution for the number of instances (individuals) among these types. For component diversity, the focus will be on measuring the distances between the generated protocols.

6 Operation support to IoT system Security, Privacy and Resilience

This section describes the support intended to be offered by ENACT to IoT system operation with respect to ensuring a secure, resilient and privacy-respectful behaviour. The support includes tools to both continuously monitor the trustworthiness level of the system and detect any possible incidents, and to promptly react to detected problems.

While monitoring support will be offered by the so-called *Security and Privacy Monitoring and Control Enabler*, resilience support will be provided by the *Robustness and resilience Enabler*, also named *Diversifier*. In the following, Section 6.1 and Section 6.2 describe both enablers respectively.

6.1 Security and Privacy Monitoring and Control Enabler

The smart preventive security mechanisms in ENACT will include the continuous monitoring of (i) security metrics and (ii) the context with the objective to early identify anomalies and attacks and promptly trigger reactive security measures. The measuring of security metrics and privacy metrics is covered in ENACT by the Security and Privacy Monitoring and Control Enabler in WP4. (Note that the context monitoring will be the focus of Context Monitoring and Actuation Conflict Management Enabler in WP3.)

In Figure 5 the main functional components envisaged for the Security and Privacy Monitoring and Control Enabler are depicted. These are:

- *Data collector*: devoted to acquiring the data from the information sources by means of distributed probes deployed in the different layers of the SIS. The data collected will range from network traffic packages to device and edge logs that will be stored in a Raw data repository.
- *Data pre-processor*: Usually it would be necessary to classify data, unify data formats, and normalise data. The process would usually involve adding required metadata (e.g., semantic tags) and performing some additional operation (e.g., filtering).
- *SIEM (Security Information and Event Management)*: This component is responsible for the specific detection processing. It performs the real-time analysis based on correlation of pre-processed network traffic data as well as logs and security alerts generated by devices and applications in the system. The SIEM includes first line root-cause analysis features and is able to generate alerts if analysis indicates a potential security or privacy issue. The SIEM offers an easy-to-use and friendly visualisation tool to SIS operators so as full situational awareness is possible.
- *Reaction manager*: The reaction manager is in charge of analysing the detected incident so as to decide the best reaction strategy to recover to a secure or privacy-respectful status in the SIS. The manager will rely on existing pre-defined reaction models and rules. In cases when reaction automation is possible, this module will call other ENACT tools to enforce specific controls that keep information secure, e.g., activate the Context-aware Access Control, request a deployment of a patched software version in a specific component, etc.

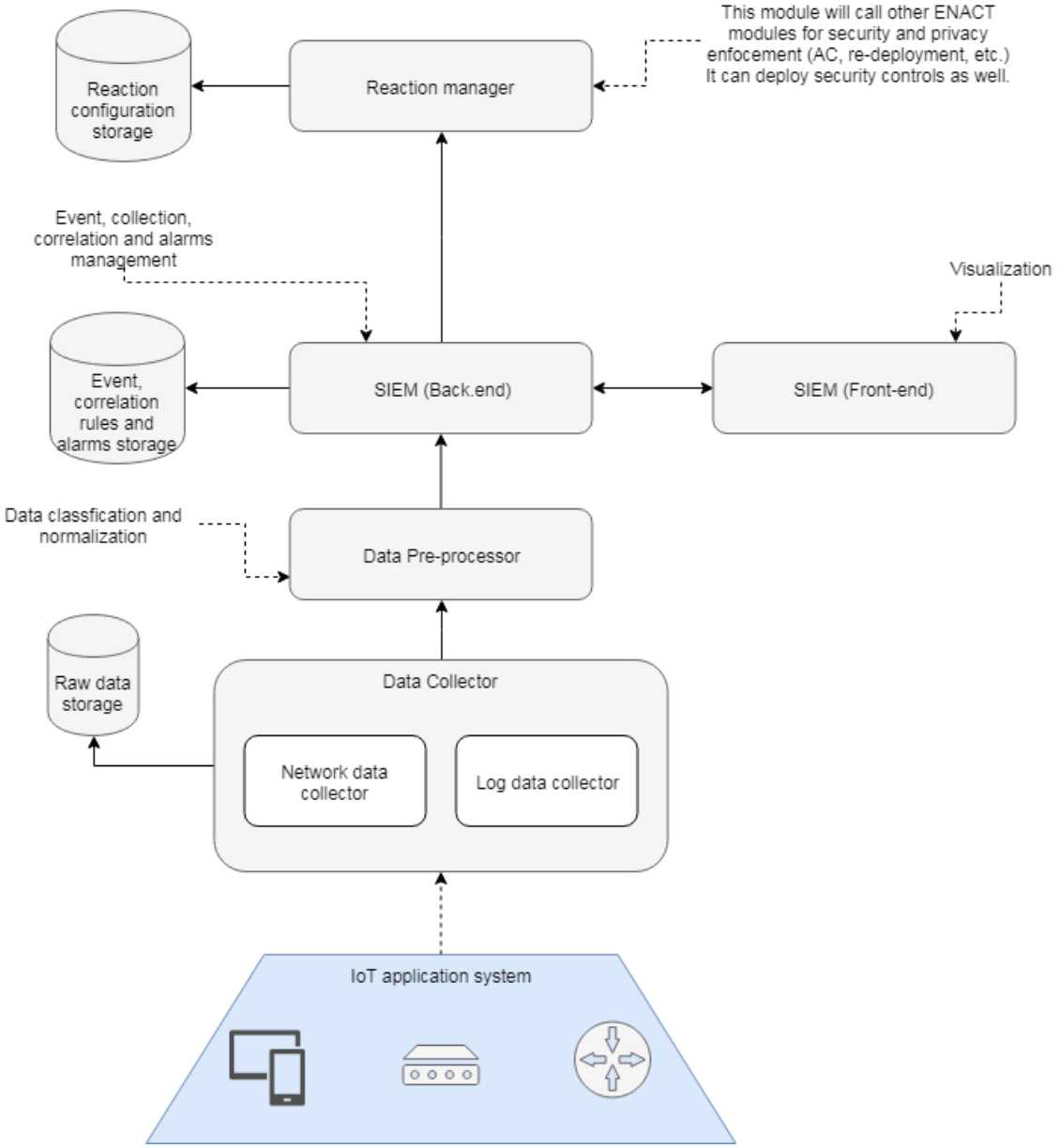

Figure 5 – ENACT Security & Privacy Monitoring Enabler

6.1.1 Monitoring mechanisms

The Security and Privacy monitoring tool will provide mechanisms to monitor end-to-end the security, privacy of a smart IoT system, with a two-fold purpose:

- To detect malicious activity and identify attacks as early as possible by combining multiple information from the different layers and controls in the SIS.
- To check the effectiveness of the security and privacy mechanisms used at run-time, enabling the mechanisms to be used in a cost-effective way, and speeding up the process of demonstrating compliance with relevant data protection standards.

The goal in ENACT will be to support the use of contingency plans to provide continuous protection. To this aim, we will leverage open source solutions, particularly for network and system levels monitoring, while new innovative solutions will be developed for the application level security and privacy assessment.

Monitoring can determine whether the necessary protections are correctly in force in the IoT system once it has been deployed. These techniques will include monitoring system components to check that classical security controls such as authentication, authorization and encryption are effectively used and also monitoring access to data to detect potential violations. These tools will support 'run-time' security and GDPR compliance monitoring.

The most challenging task will be to identify new threats or zero-day vulnerabilities by correlating data from distributed sources and probes at the network, application, cloud and IoT environment level. To this aim, ENACT will rely on mechanisms for the continuous monitoring of all network activity, application usage, users and threats in order to detect anomalies and events that may be the symptoms of new cyberattacks. Innovative attack pattern extraction techniques may be required by correlating events to identify hidden attack patterns and trends. The main challenges of the correlation strive in the variety and amount of data and logs from devices and elements in IoT system as well as in the large heterogeneity among them.

A set of relevant metrics will be defined and notifications will be raised when the monitored metrics deviated from the normal (risk under control) behaviour. The enabler will include mechanisms and tools to support the user data awareness and control in form of intelligent notification able to provide insights on what is actually the security issue in the IoT environment.

6.1.2 Reaction mechanisms

The Reaction manager will be able to decide the most appropriate reaction measures to recover from the detected incident and therefore, it will need to be able to orchestrate a number of possible reactions. The reaction measures include the activation of security and privacy (data protection) controls and the invocation of other ENACT tools for other types of adaptation, e.g., for diversity purposes.

The DevOps approach enables to deploy features into production quickly and to detect and correct problems when they occur, without disrupting other services, thanks to its continuous integration, continuous testing and continuous deployment philosophy and accompanying tools.

It is often believed that current DevOps already include security concerns in the workflow but reality shows that security is often overlooked with the rush to bring the product out in the market [106]. ENACT promotes to include security experts and team members in the development and operation of applications that later on will be deployed. Current DevOps ignore on one hand the inclusion of security experts as a part of the stable development and deployment team, and on the other hand, available DevOps focus on continuous testing, continuous integration and continuous testing overlooking security patterns and mechanisms [106], such as the ones to be developed in ENACT. To successfully hook security and privacy aspects into classic DevOps development processes, the key is to add threat identification, risk assessment, and monitoring as early as necessary/possible, as ENACT intends to do, so as reaction measures can be decided and enforced as soon as possible.

6.1.3 Context-Aware Access Control mechanisms

The Context-aware Access Control tool with the Security and Privacy Monitoring and Control Enabler will provide an Authorization mechanism that will issue access tokens to the connected objects after successfully authenticating their owner and obtaining authorization. This Authorization mechanism will use the OAuth2 protocol, which provides authorization delegation mechanism. Following this protocol, an object will be able to access a backend API by using an access token containing the list of scopes and claims that an authenticated user has consented for this object to access. An Access token contains an authentication proof and the list of consented scopes and claims to access the asked resource.

This Authorization mechanism may be coupled with contextual information to adapt the access authorizations according to them (for example to make certain information more widely available in some urgent case).

The Context-aware Access Control tool will provide access tokens that allow a Reverse Proxy working as an API Gateway to control the access to applications and APIs. The scopes and claims contained in the access tokens are used to restrict accesses to the backend server APIs to a consented set of resources.

The Authorization mechanism could be coupled to a multi-level, multi-factor Authentication Server that provides strong authentications mechanisms to the users. This mechanism mitigates the level of authentication required depending on the user's environment context and an external context. The risk is computed either statically, depending on a defined configuration, or dynamically by using a REST API to dialog with an external decision engine. The transmitted input is the session context. Depending on the evaluated risk of the user's session, the level of the required authentication will be leveled up, or, if the risk is too high, the connection will be refused.

These features may be architectured as shown in the global schema of Figure 6.

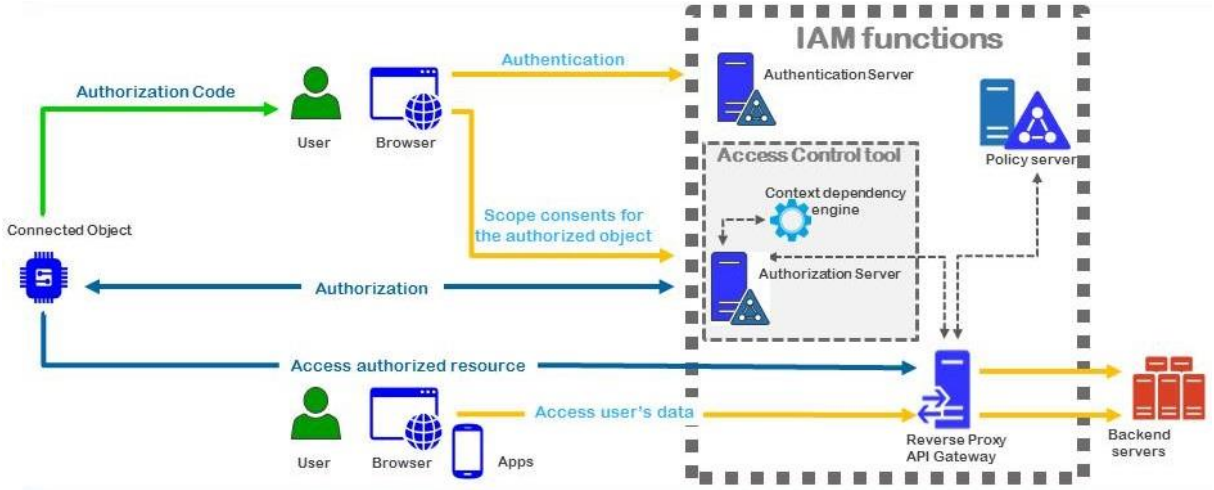

Figure 6 – ENACT Context-Aware AC architecture

Description of the **Context-aware Access Control** mechanisms:

When a connected object has to send information to a backend server, the access to the data managed by the backend server must be controlled to ensure that the connected object handles only the data of the person who owns it, and to ensure that only authorized persons will consult this information. This is ensured by the following mechanisms:

- When initialized, the connected object asks for an authorization to the Authorization Server, with a list of scopes (i.e., information managed by this device) it wants to access on the backend server. The Authorization Server provides it with an authorization code.

- This authorization code is transmitted to the device owner.

- The user (owner of the device) authenticates (on the Authentication server) and enters the authorization code which identifies the device. Then he accepts or declines the scopes requested by the device. The Authorization server establishes a link between the device and the user, and emits an access token to the device.

- The device is then ready to emit data to the backend server in a controlled way, by addressing the Reverse Proxy (API Gateway) with the obtained access token.

- The Reverse Proxy asks the Authorization server for the token verification, in order to use the consented scopes to restrict accesses to the backend server APIs.

- The user accesses the data produced by the connected object also in a controlled way through the reverse proxy.

6.1.4 Security and Privacy adaptation mechanisms in IoT platform

The ITS and Smart building use cases in ENACT both use SOFIA based IoT platforms (SOFIA 2.0 and SMOOL platforms, respectively). Based on the communication protocols that SOFIA can manage, it is possible to include monitoring and reaction (notification) mechanisms by analysing the security on the communications among the things and the IoT platform for detecting security vulnerabilities and anomalies.

In ENACT a set of security capabilities for SOFIA have been designed and are currently being developed. More concretely, a KP-client for Security in SMOOL is under development. Such KP-client will act as proxy between KPs and allow processing different data and metadata sent by KPs (e.g., sensors). Together with this Security KP-client, SMOOL server will also need to be extended in order to effectively enforce the controls.

For example, as shown in next figure, it is possible to add a check of timestamp validity before allowing communications between KPs. Other possible controls in the data sent by a KP can include specific allowed vendor, only real-time data, check of existence of mandatory fields in the message, etc.

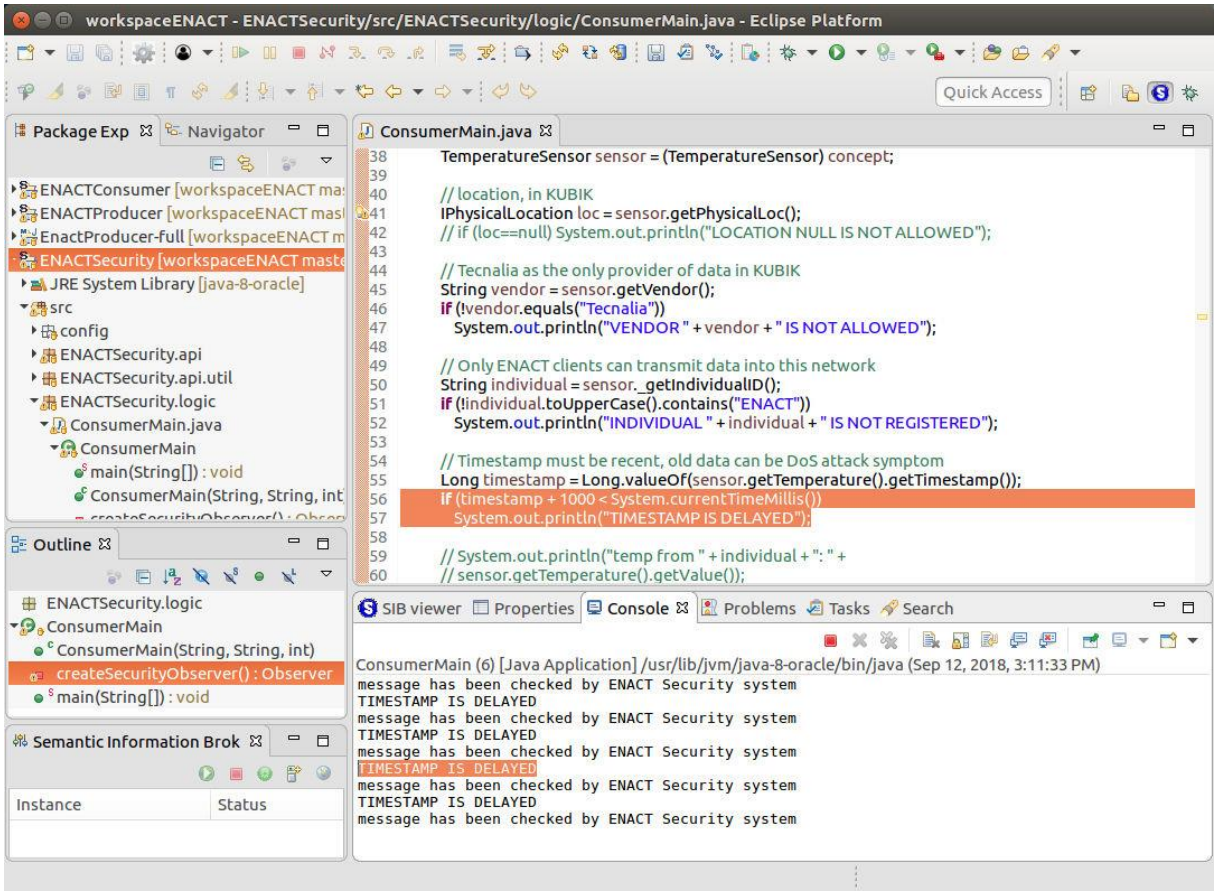

Figure 7 – ENACT Generic Security KP-client in SMOOL platform

6.1.5 Other control mechanisms

Besides the mechanisms described above, as part of a well-designed reaction strategy per identified threat, the enforcement of other security mechanisms is also possible:

- Vulnerability scanning solutions: This security control relies in providing the necessary Software Vulnerability Assessment (SVA) tools for the IoT application to be protected. The offered SVA tools can detect and upgrade the vulnerabilities or fix the misconfigurations in the specified software packages deployed usually in a web or cloud container. Examples of this type of solutions are the open source OpenVAS [107] and the SVA tool in SPECS platform [108].
- Backup solutions: This security control relies in providing the necessary backup capability to the IoT application to ensure its resilience and recovery readiness. The backup features may include compression, encryption, source file filters, delta backup, archive merges, as-of-date recovery, reports, etc. Examples of well-known open source backup software are: Zmanda [109] which enables to backup data from live applications and databases directly to a storage cloud (Zmanda Cloud Backup (ZCB) backs up a Windows server and live applications such as Microsoft Exchange and SQL Server to Amazon S3), Areca Backup [110], and Bacula [111]. A comprehensive collection of available open source backup solutions can be found in [112].

Although not identified as main priority, it is currently under study the possibility of including these or similar enforcement mechanisms in the ENACT framework.

6.2 Robustness and resilience Enabler – Diversifier

The Robustness and resilience Enabler in ENACT will only include one tool, the Diversifier, which closely work with the Orchestration and Continuous Deployment Enabler, a.k.a. GeneSIS framework in ENACT (see deliverable D2.1 for more information on this enabler). The Diversifier will be used at operation for adapting the SIS to address diversity requirements whenever needed.

6.2.1 Diversity-aware adaptation mechanisms

At run-time, the ENACT Diversifier relies on the run-time adaptation capability provided by the ENACT deployment and orchestration tool to perform diversity-aware adaptation. After the automatic diversity generation at design time, all the generated components and specifications, as well as the alternative third-party components, are all registered in assets repositories, such as GitHub and Docker Hub. At run-time, the diversity-aware adaptation engine switches the current system into an alternative one by invoking the deployment tool with the new deployment model. The deployment tool will execute the new model, download the required components from the repositories and configure the them into an integrated system. In this process, the diversifier is only in charge of the decision on when to switch to a new architecture, and to which of the alternative architecture. The diversifier adopts different adaptation strategies to make such decisions, depending on the use cases and the requirements.

Based on the current use cases in the ENACT projects, we foresee the following two types of strategies:

1. **Diversity-aware adaptation for configuration testing:**

   The configuration of a gateway denotes the software and libraries deployed in the gateway, their versions and the parameters set on them. Since the end users may end up using different configurations, the testing should cover at least the representative configurations. In the continuous integration pipeline, if the “integration testing” or “configuration testing” flag is on (which means that the system is under a daily or weekly building, rather than unit testing for developers), the diversifier will start a loop of testing and in each iteration, the diversifier will take one of the generated deployment specification, employ the deployment engine to install the testing hardware accordingly, and run the integration test suites on the gateway.

2. **Diversity for recovery**:

   After the Gateway is released, the ENACT diversifier will automatically transform the Gateway from the current configuration to an alternative one, when the system is under exceptional conditions, such as system downtime, bad performance (too long response time), extreme loads, frequent errors, etc. The type of exceptional conditions to monitor, as well as the threshold to trigger the adaptation, are defined according to the use case. The new configuration to switch to can be randomly selected, or based on experience from previous adaptations.

7 Conclusions

The goal of WP4 in ENACT is to provide support to define and ensure the secure, resilient and privacy-aware behaviour of smart IoT systems. The work package will deal with support to development and operations phase of the DevOps cycle. At development phase the support will mainly include system security, privacy and resilience (diversity) requirements specification mechanisms together with the associated controls specification. At operations phase, the support will be focused on continuous monitoring of possible security and privacy incidents and attacks to the IoT system as well as early reaction to them.

The main focus of this document is the description of the state of the art in security, privacy and resilience (diversity) solutions for IoT systems and the initial study of the security and privacy requirements derived from the analysis of the use cases of the project. In addition, the document describes the initial plans for developing mechanisms and tools that will be integrated in the overall ENACT framework to support both developers and operators addressing security, privacy and resilience aspects of SIS.

The development mechanisms that will be developed by WP4 will mainly support SIS developers in improving the design of SIS by including the necessary security, privacy and resilience information to ensure risks are minimised in the deployed SIS.

The operational mechanisms by WP4 will be deployed in different layers of the SIS, ranging from the control of the use of encryption in the communications to access control mechanisms at both IoT platform and application levels.

WP4 will develop the detailed design and implementation of the following enablers: i) a Security and Privacy Monitoring and Control enabler, which includes an advanced context-aware access control mechanism and ii) a Diversifier to be able to analyse diversity requirements of SIS and request software variants deployments when needed.

8 References

[1] Karabacak, B. and I. Sogukpinar, ISRAM: information security risk analysis method. Computers & Security, 2005. 24(2): p. 147-159.
[2] Alberts, C.J. and A. Dorofee, Managing information security risks: the OCTAVE approach. 2002: Addison-Wesley Longman Publishing Co., Inc.
[3] Lund, M.S., B. Solhaug, and K. Stølen, Model-driven risk analysis: the CORAS approach. 2010: Springer.
[4] Rao, L.M. and S. Firdose, *Study of Existing Risk Management Models and Prior Research Contribution.* Adarsh Journal of Information Technology, 2016. **4**(1): p. 10-20.
[5] Borgia, Eleonora. "The Internet of Things vision: Key features, applications and open issues." *Computer Communications* 54 (2014): 1-31.
[6] Jing, Qi, et al. "Security of the Internet of Things: perspectives and challenges." Wireless Networks 20.8 (2014): 2481-2501.
[7] Wu, Miao, et al. "Research on the architecture of Internet of things." Advanced Computer Theory and Engineering (ICACTE), 2010 3rd International Conference on. Vol. 5. IEEE, 2010.
[8] Gama, Kiev, Lionel Touseau, and Didier Donsez. "Combining heterogeneous service technologies for building an Internet of Things middleware." Computer Communications 35.4 (2012): 405-417.
[9] Granjal, J., Monteiro, E., & Silva, J. S. (2015). Security for the internet of things: a survey of existing protocols and open research issues. IEEE Communications Surveys & Tutorials, 17(3), 1294-1312.
[10] Bormann, C., Castellani, A. P., & Shelby, Z. (2012). Coap: An application protocol for billions of tiny internet nodes. IEEE Internet Computing, 16(2), 62-67.
[11] MQTT Version 3.1.1. OASIS Standard. Latest version: http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/mqtt-v3.1.1.html. (Accessed October 2018)
[12] XMPP, Extensible Messaging and Presence Protocol. Online: https://xmpp.org/ (Accessed October 2018)
[13] Cirani, S., Picone, M., Gonizzi, P., Veltri, L., & Ferrari, G. (2015). Iot-oas: An oauth-based authorization service architecture for secure services in iot scenarios. IEEE sensors journal, 15(2), 1224-1234.
[14] Domenech, M. C., Comunello, E., & Wangham, M. S. (2014, October). Identity management in e-Health: A case study of web of things application using OpenID connect. In e-Health Networking, Applications and Services (Healthcom), 2014 IEEE 16th International Conference on (pp. 219-224). IEEE.
[15] Lee, J. J., Hong, Y. S., & Lee, K. Y. (2015, January). An Authentication Scheme Based on Elliptic Curve Cryptosystem and OpenID in the Internet of Things. In Proceedings of the International Conference on Security and Management (SAM) (p. 192). The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp).
[16] E. Rescorla and N. Modadugu, DTLS: Datagram Transport Layer Security, RFC 4347, 2006.
[17] Shahid Raza, Hossein Shafagh, Kasun Hewage, Rene Hummen, and Thiemo Voigt. Lithe: Lightweight secure CoAP for the internet of things. IEEE Sensors Journal, 13(10):3711– 3720, 2013.
[18] Keoh, S., Garcia-Morchon, O., Kumar, S., & Dijk, S. (2012). DTLS-based Multicast Security for Low-Power and Lossy Networks (LLNs). work-in-progress.
[19] Roman, Rodrigo, et al. "Key management systems for sensor networks in the context of the Internet of Things." Computers & Electrical Engineering 37.2 (2011): 147-159.
[20] Hong, Ning. "A security framework for the internet of things based on public key infrastructure." Advanced Materials Research. Vol. 671. Trans Tech Publications, 2013.
[21] Eisenbarth, T., & Kumar, S. (2007). A survey of lightweight-cryptography implementations. IEEE Design & Test of Computers, 24(6).

[22] Koblitz, N. (1987). Elliptic curve cryptosystems. Mathematics of computation, 48(177), 203-209.
[23] Zhao, Yan Ling. "Research on data security technology in internet of things." Applied Mechanics and Materials. Vol. 433. Trans Tech Publications, 2013.
[24] Kothmayr, Thomas, et al. "DTLS based security and two-way authentication for the Internet of Things." Ad Hoc Networks 11.8 (2013): 2710-2723.
[25] Platform Industry 4.0, Federal Ministry for Economic Affairs and Energy (BMWi). "Technical Overview: Secure Identities". April, 2016.
[26] "Internet of Things Top Ten," Open Web Application Security Project, 2014; Online: www.owasp.org/images/7/71/Internet_of_Things_Top_Ten_2014-OWASP.pdf. (Accessed October 2018)
[27] Constantinos Kolias, Angelos Stavrou, Jeffrey M. Voas, Irena V. Bojanova, David R. Kuhn. "Learning Internet of Things Security "Hands-on"", 2016. Online: https://dx.doi.org/10.1109/MSP.2016.4 (Accessed October 2018)
[28] Singh, J., Pasquier, T., Bacon, J., Ko, H., & Eyers, D. (2016). Twenty security considerations for cloud-supported Internet of Things. IEEE Internet of Things Journal, 3(3), 269-284.
[29] IEC Whitepaper: "IoT 2020: Smart and secure IoT platform". Online: http://www.iec.ch/whitepaper/pdf/iecWP-loT2020-LR.pdf
[30] Dalipi, F., & Yayilgan, S. Y. (2016, August). Security and Privacy Considerations for IoT Application on Smart Grids: Survey and Research Challenges. In Future Internet of Things and Cloud Workshops (FiCloudW), IEEE International Conference on (pp. 63-68). IEEE.
[31] Hossain, M. M., Fotouhi, M., & Hasan, R. (2015, June). Towards an analysis of security issues, challenges, and open problems in the internet of things. In Services (SERVICES), 2015 IEEE World Congress on (pp. 21-28). IEEE.
[32] Mendez, D. M., Papapanagiotou, I., & Yang, B. (2017). Internet of things: Survey on security and privacy. arXiv preprint arXiv:1707.01879.
[33] Security and Resilience of Smart Home Environments , ENISA, December 2015, Online: https://www.enisa.europa.eu/publications/security-resilience-good-practices/at_download/fullReport. (Accessed October 2018)
[34] Baseline Security Recommendations for IoT, ENISA, November 2017. Online: https://www.enisa.europa.eu/publications/baseline-security-recommendations-for-iot/at_download/fullReport . (Accessed October 2018)
[35] Towards secure convergence of Cloud and IoT, ENISA, September 2018. Online: https://www.enisa.europa.eu/publications/towards-secure-convergence-of-cloud-and-iot/at_download/fullReport (Accessed October 2018)
[36] LI, Juan; OUEDRAOGO11, Wendpanga Francis; BIENNIER, Frédérique. Multi-Cloud Governance Service based on Model Driven Policy Generation.
[37] CLAVEL, Manuel, et al. Model-driven security in practice: An industrial experience. En Model Driven Architecture–Foundations and Applications. Springer Berlin Heidelberg, 2008. p. 326-337.
[38] WOLTER, Christian, et al. Model-driven business process security requirement specification. Journal of Systems Architecture, 2009, vol. 55, no 4, p. 211-223.
[39] Model-based provisioning and deployment of cloud-based systems. CloudML project. Available at: http://cloudml.org
[40] FERRY, Nicolas, et al. Towards model-driven provisioning, deployment, monitoring, and adaptation of multi-cloud systems. In CLOUD 2013: IEEE 6th International Conference on Cloud Computing. 2013. p. 887-894.
[41] OMG Model-Driven Architecture. URL: http://www.omg.org/mda/.
[42] PaaSage consortium. D2.1.2 CloudML Implementation Documentation. April 2014. Available at: http://www.paasage.eu/images/documents/paasage_d2.1.2_final.pdf (Accessed October 2018)
[43] PaaSage EU FP7 project. Available at: http://www.paasage.eu/

[44] CAMEL Documentation v2015.9 by PaaSage EU Project. Available at: http://camel-dsl.org/documentation/ (Accessed October 2018)
[45] Clément Quinton, Daniel Romero and Laurence Duchien. 'Cardinality-based feature models with constraints: a pragmatic approach'. In: SPLC 2013: 17th International Software Product Line Conference. Ed. by Tomoji Kishi, Stan Jarzabek and Stefania Gnesi. ACM, 2013, pp. 162–166. ISBN: 978-1-4503-1968-3. DOI: 10.1145/2491627.2491638.
[46] Keith Jeffery, Nikos Houssos, Brigitte Jörg and Anne Asserson. 'Research information management: the CERIF approach'. In: IJMSO 9.1 (2014), pp. 5–14. DOI: 10.1504/IJMSO.2014.059142.
[47] Kyriakos Kritikos, Jörg Domaschka and Alessandro Rossini. "SRL: A Scalability Rule Language for Multi-Cloud Environments". In: Cloud-Com 2014: 6th IEEE International Conference on Cloud Computing Technology and Science. Ed. by Juan E. Guerrero. IEEE Computer Society, 2014, pp. 1–9. ISBN: 978-1-4799-4093-6. DOI: 10.1109/CloudCom.2014.170.
[48] Karniavoura, F., Papaioannou, A., & Magoutis, K. (2015). C2C: An Automated Deployment Framework for Distributed Applications on Multi-Clouds. In Proceedings of 9th Symposium and Summer School On Service-Oriented Computing, Hersonissos, Crete, Greece.
[49] MUSA H2020 project, https://www.musa-project.eu (Accessed October 2018)
[50] Rios, E., Iturbe, E., & Palacios, M. C. (2017). Self-healing Multi-Cloud Application Modelling.
[51] Topology and Orchestration Specification for Cloud Applications Standard. TOSCA standard by OASIS. Available at http://docs.oasis-open.org/tosca/TOSCA/v1.0/TOSCA-v1.0.html
[52] Teixeira, T., et al. Service oriented middleware for the internet of things: a perspective. in European Conference on a Service-Based Internet. 2011. Springer.
[53] ISO: 'ISO/IEC 20000-1:2011 information technology - service management - part 1: Service management system requirements', 2011.
[54] National Institute of Standards and Technology (NIST), 'Security and Privacy Controls for Information Systems and Organizations'. NIST SP-800-53, revison 5 Draft.
[55] Rak, M.: 'Security assurance of (multi-) cloud application with security SLA composition'. Proc. Int. Conf. on Green, Pervasive, and Cloud Computing, Springer (2017) pp. 786-799.
[56] Casola, V., Benedictis, A.D., Rak, M., Villano, U.: 'A security metric catalogue for cloud applications'. Proc. Int. Conf. on Complex, Intelligent, and Software Intensive Systems (CISIS), July 2017, pp. 854-863.
[57] Learning Internet of Things Security "Hands-on" C. Kolias, A.Stavrou, J.M. Voas, I.V. Bojanova, D.R. Kuhn, 2016. Online: https://dx.doi.org/10.1109/MSP.2016.4
[58] Cvitic, I. and Vujic, M. and Husnjak, S. (2015), "Classification of Security Risks in the IoT Environment", 26th DAAAM International Symposium on Intelligent Manufacturing and Automation, p731-740
[59] Mahmud, H. and Maziar, F. and Ragib H. (2015), "Towards an Analysis of Security Issues, Challenges, and Open Problems in the Internet of Things".
[60] J. Jacobs, B. Rudis, "Data-Driven Security: Analysis, Visualization and Dashboards", John Wiley & Sons, Inc., April 2014.
[61] G. Hohpe, "Programming Without a Call Stack – Event-driven Architectures", 2006
[62] http://kafka.apache.org (Accessed October 2018)
[63] http://www.rabbitmq.com (Accessed October 2018)
[64] D. Chou, "Using Events in Highly Distributed Architectures", The Architecture Journal, Microsoft Corporation, no. 17, October 2008.
[65] https://www.ossec.net/ (Accessed October 2018)https://www.alienvault.com/products/ossim (Accessed October 2018)
[66] Zbakh, M. et al. 2015. A multi-criteria analysis of intrusion detection architectures in cloud environments. 2015 international conference on cloud technologies and applications (cloudTech) (2015), 1–9.

[67] Casola, V., De Benedictis, A., & Rak, M. (2015, August). Security monitoring in the cloud: an SLA-based approach. In Availability, Reliability and Security (ARES), 2015 10th International Conference on (pp. 749-755). IEEE.
[68] Rios, E., Iturbe, E., Mallouli, W., & Rak, M. (2017, October). Dynamic security assurance in multi-cloud DevOps. In *Communications and Network Security (CNS), 2017 IEEE Conference on* (pp. 467-475). IEEE.
[69] NISP WG1 – Risk Management Best Practice -Final Recommendations, Issue 1 280414, 2014.
[70] Casola, V., De Benedictis, A., Eraşcu, M., Modic, J., & Rak, M. (2017). Automatically enforcing security slas in the cloud. *IEEE Transactions on Services Computing*, *10*(5), 741-755.
[71] https://prismacloud.eu/toolbox/ (Accessed October 2018)
[72] Mahmud, H. and Maziar, F. and Ragib H. (2015), "Towards an Analysis of Security Issues, Challenges, and Open Problems in the Internet of Things".
[73] Cvitic, I. and Vujic, M. and Husnjak, S. (2015), "Classification of Security Risks in the IoT Environment", 26th DAAAM International Symposium on Intelligent Manufacturing and Automation, p731-740.
[74] Fall, D. and Okuda, T. and Kadobayashi, Y. and Yamaguchi, S. (2016), "Risk Adaptive Authorization Mechanism (RAdAM) for Cloud Computing, Journal of Information Processing, Vol 24 No.2, p371-380.
[75] Farooq, M.U. and Waseem, M. and Khairi, A. and Mazhar, S. (2015), "A critical Analysis on the Security Concerns of Internet of Things (IoT)", International Journal of Computer Applications, Volume 111 No.7.
[76] Jagadamba, G. and Sathish Babu, B. (2016), "Adaptive Security Schemes based on Context and Trust for Ubiquitous Computing Environment: A Comprehensive Survey", Indian Journal of Science & Technology, Vol 9 (48).
[77] Habib, K. and Leister, W. (2015), "Context-Aware Authentication for the Internet of Things", ICAS 2015: The Eleventh International Conference on Autonomic and Autonomous Systems, p134-139.
[78] Dankar, F. and Badji, R. (2017), "A risk-based framework for biomedical data sharing", Journal of Biomedical Informatics, Vol 66, p231-240.
[79] Fernandes, E. and Jung, J. and Prakash, A. (2016), "Security Analysis of Emerging Smart Home Applications".
[80] Hiller, J. and Russel, R. (2017),"Privacy in Crises: The NIST Privacy Framework", Journal of Contingencies and Crisis Management, Volume 25 Number 1, p31-38.
[81] Oxford dictionary: https://en.oxforddictionaries.com/definition/resilience
[82] https://cabforward.com/the-difference-between-reliable-and-resilient-software/
[83] A short summary at: Benson, K., 2015, March. Enabling resilience in the Internet of Things. In Pervasive Computing and Communication Workshops (PerCom Workshops), 2015 IEEE International Conference on (pp. 230-232). IEEE.
[84] K. Mikhaylov, J. Petäjäjärvi, M. Mäkeläinen, A. Paatelma, and T. Hänninen, "Demo: Modular multi-radio wireless sensor platform for IoT trials with plug&play module connection," in Proc. ACM Int. Conf. Mobile Comput. Netw., Paris, France, 2015, pp. 188–189.
[85] A. Munir, A. Gordon-Ross, and S. Ranka, "Multi-core embedded wireless sensor networks: Architecture and applications," IEEE Trans. Parallel Distrib. Syst., vol. 25, no. 6, pp. 1553–1562, Jun. 2014.
[86] X. Xie, H. Chen, and H. Wu, "Bargain-based stimulation mechanism for selfish mobile nodes in participatory sensing network," in Proc. IEEE Sensor Mesh Ad Hoc Commun. Netw. (SECON), Rome, Italy, 2009, pp. 1–9.
[87] Oteafy, S.M. and Hassanein, H.S., 2017. Resilient IoT architectures over dynamic sensor networks with adaptive components. IEEE Internet of Things Journal, 4(2), pp.474-483.

[88] C. Liang and F. R. Yu, "Wireless network virtualization: A survey, some research issues and challenges," IEEE Commun. Surveys Tuts., vol. 17, no. 1, pp. 358–380, 1st Quart., 2015.
[89] Delic, K.A., 2016. On resilience of iot systems: The internet of things (ubiquity symposium). Ubiquity, 2016(February), p.1.
[90] Kevin Shear McCann. 2000. The diversity–stability debate. Nature 405, 6783, 228–233.
[91] The N-version approach to fault-tolerant software. IEEE Transactions on Software Engineering 11, 12, 1491–1501.
[92] Design diversity and the immune system paradigm: Cornerstones for information system survivability. In Proceedings of the 3rd Information Survivability Workshop (ISW'00).
[93] Alex X. Liu and Mohamed G. Gouda. 2008. Diverse firewall design. IEEE Transactions on Parallel and Distributed Systems 19, 9, 1237–1251.
[94] Stephanie Forrest, Anil Somayaji, and David H. Ackley. 1997. Building diverse computer systems. In Proceedings of the 6th Workshop on Hot Topics in Operating Systems (HOTOS'97). IEEE, Los Alamitos, CA, 67. http://dl.acm.org/citation.cfm?id=822075.822408
[95] Elena Gabriela Barrantes, David H. Ackley, Trek S. Palmer, Darko Stefanovic, and Dino Dai Zovi. 2003. Randomized Instruction Set Emulation to Disrupt Binary Code Injection Attacks. Technical Report TRCS-2003-10. University of New Mexico.
[96] Todd Jackson. 2012. On the Design, Implications, and Effects of Implementing Software Diversity for Security. Ph.D. Dissertation. University of California, Irvine.
[97] Sandeep Bhatkar, Daniel C. DuVarney, and Ron Sekar. 2003. Address obfuscation: An efficient approach to combat a broad range of memory error exploits. In Proceedings of the USENIX Security Symposium.
[98] Matti A. Hiltunen, Richard D. Schlichting, Carlos A. Ugarte, and Gary T. Wong. 2000. Survivability through customization and adaptability: The Cactus approach. In Proceedings of the DARPA Information Survivability Conference and Exposition (DISCEX'00), Vol. 1. 294–307.
[99] Juan Caballero, Theocharis Kampouris, Dawn Song, and Jia Wang. 2008. Would diversity really increase the robustness of the routing infrastructure against software defects? In Proceedings of the 16th Annual Network and Distributed System Security Symposium (NDSS'08).
[100] Eric Totel, Fr´ed´eric Majorczyk, and Ludovic Me. 2006. COTS diversity based intrusion detection and application to Web servers. In Recent Advances in Intrusion Detection. Springer, 43–62.
[101] Rong Wang, Feiyi Wang, and Gregory T. Byrd. 2003. Design and implementation of Acceptance Monitor for building intrusion tolerant systems. Software: Practice and Experience 33, 14, 1399–1417.
[102] Noguero, A., Rego, A., & Schuster, S. Towards a Smart Applications Development Framework. Social Media and Publicity, 27.
[103] http://www.escudocloud.eu/index.php/menu-results/menu-software (Accessed October 2018)
[104] http://clarussecure.eu/clarus-proxy-high-level-architecture (Accessed October 2018)
[105] Deliverable D3.3 of MUSA project, https://www.musa-project.eu/documents2/d33-final-security-based-discovery-and-composition-mechanisms-and-tools (Accessed October 2018)
[106] https://project-shield.eu/Tools/Details/4 (Accessed October 2018)
[107] OpenVAS. Open Source vulnerability scanner and manager. Available at: http://openvas.org/ (Accessed October 2018)
[108] The SPECS project deliverable D4.2.2. Available at: http://www.specs-project.eu/publications/public-deliverables/d4-2-2/ (Accessed October 2018)
[109] https://www.zmanda.com/company.html (Accessed October 2018)
[110] http://www.areca-backup.org/ (Accessed October 2018)
[111] https://blog.bacula.org/ (Accessed October 2018)

[112] http://www.enterprisestorageforum.com/backup-recovery/open-source-storage-49-tools-for-backup-and-recovery.html (Accessed October 2018)