
| | |
|---|---|
| *Title:* | *ENACT DevOps Framework – Final version .* |
| *Authors:* | *Alexander Palm (UDE), Thorsten Weyer (UDE), Jacek Dominiak (Beawre), Nicolas Ferry (CNRS), Hui Song (SINTEF), Rustem Dautov (SINTEF), Phu Nguyen (SINTEF), Franck Dechavanne (CNRS), Thibaut Gonnin (CNRS), Stéphane Lavirotte (CNRS), Gérald Rocher (CNRS), Jean-Yves Tigli (CNRS), Luong NGUYEN (Montimage),Vinh Hoa La (Montimage), Wissam Mallouli (Montimage), Edgardo Montes de Oca (Montimage), Anne Gallon (Evidian), Jean-Christophe Durieu (Evidian), Samuel Mamuye (Evidian), Olivier Verdun (Evidian), Elena González (Beawre), Victor Muntés (Beawre), Eider Iturbe (Tecnalia), Saturnino Martinez (Tecnalia), Angel Rego (Tecnalia) and Erkuden Rios (Tecnalia)* |
| *Editor:* | *Jacek Dominiak (Beawre)* |
| *Reviewers:* | *Modris Greitans (EDI), Uģis Grīnbergs (BOSC)* |
| *Identifier:* | *Deliverable # D5.4* |
| *Nature:* | *Software* |
| *Date:* | *01 March 2021* |
| *Status:* | *v1.0* |
| *Diss. level:* | *Public* |

**Executive Summary**

This deliverable showcases the final release of the ENACT framework, along with the API's descriptions available for the integration as well as the list of market leaders which can be integrated with the enablers of the framework.

Members of the ENACT consortium:

| | |
|---|---|
| SINTEF AS | Norway |
| BEAWRE DIGITAL SL | Spain |
| EVIDIAN SA | France |
| INDRA Sistemas SA | Spain |
| Fundacion Tecnalia Research & Innovation | Spain |
| TellU AS | Norway |
| Centre National de la Recherche Scientifique | France |
| Universitaet Duisburg-Essen | Germany |
| MONTIMAGE | France |
| 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 |
|---|---|---|---|
| | V0.1 | Jacek Dominiak | Table of content |
| 15/12/2020 | V0.2 | Jacek Dominiak, Nicolas Ferry | Table of content modification, include external documents in this deliverable |
| 16/12/2020 | V0.3 | Stéphane Lavirotte, Gérald Rocher, Jean-Yves Tigli | Contribute to ACM and BDA sections. |
| 17/12/2020 | V0.4 | Luong NGUYEN | Contribute to Test and Simulation section |
| 28/12/2020 | V0.5 | Jacek Dominiak | Final formatting |
| 26/01 | V10.6 | Nicolas Ferry | Add trustworthiness contribution summary |
| 02/02 | 0 | Jacek Dominiak | Add tool matching and release for internal review |
| 20/02 | V1.1 | Jacek Dominiak | Final edits based on the feedback |

Contents

# 1. Introduction

## 1.2. Context and Objectives

The objective of this deliverable is to provide support information about the final release of the ENACT framework software, including the integration and communication between the different tools of the framework, as well as their interoperability with the market leaders or other emerging tools within the same space. The deliverable builds upon the structure of D5.2 to evaluate the user stories set by each of the enabler provider and showcases the technical aspects of integration of the of the enablers of the framework targeting specifically the trustworthiness aspects of the IoT.

## 1.3. Main Achievements

As the final deliverable of the project from WP5, this document encapsulates the state of the project enablers status considered at delivery time. It is only fair to mention couple of achievements done within the project which made it exceptional.

First and foremost, ENACT was clear from the start on how each of the enablers address the parts of the DevOps cycle. Based on that note, the overall integrated architecture of the framework was done within the first year of the project. It showcased how the project outcomes are addressing the responsibilities of running an IoT system from the design phase to the monitoring phase.

Secondly, a strict set of guidelines were made in order to ensure that the project outcomes are delivered to the standards of open-source community. These guidelines set a standard for both, API and UI. It was crucial that API's exposed for the integration which can ensure that the future users of ENACT outcomes can integrate in an easy and familiar way. Similarly, from the standpoint of UI, we aimed at producing high quality UI, not only coherent from the look'n'feel perspective but also easy to use and initiative for the users with minimal domain knowledge.

Next, understandably, not only standards drive user adoption. All of the ENACT enablers where the containerization is applicable, are delivered with Dockerfiles, allowing for rapid deployment. Fast deployment is also available from the Docker Hub, where ENACT created its group.

Finally, as highlighted in the previous deliverables as well as the Section 5 of this deliverable, tools of the project are well equipped with the integration of many market leaders and accepted open-source and commercial solutions, ensuring that the project results can be noticed by the associated communities.

## 1.4. Structure of the Document

The remainder of the document is structured as follows. After a brief introduction, the extended Framework description is introduced, where the exact source code locations, documentation, tutorials and videos are listed. Within the next section, a showcase of the evolution of the use cases stories is done, to highlight the completion and/or evolution of the use cases stories from the initial state till the end of the project. Next, a list of API endpoints is showcased to ease the integration with any external tool if required. Within the section 3 a decision tree is shown in order to help the potential beneficiaries of ENACT framework decide how and when to integrate with chosen ENACT enablers. Finally, a list of detailed market leaders with which the ENACT enablers are interoperable to ease adoption is shown.

# 2. Framework description

ENACT provides an integrated DevOps Framework composed of a set of loosely coupled tools. Although not dependant on each other, these tools can be seamlessly combined, and they can easily integrate with existing IoT platform services and enablers. The exact split on each of the functional layer showcased in the Figure 1 has been described in detail within D5.2 section 2. Within this deliverable, we are presenting easy to consume consolidated version of howto's and tutorials organized within the same structure mimicking the architecture layers.

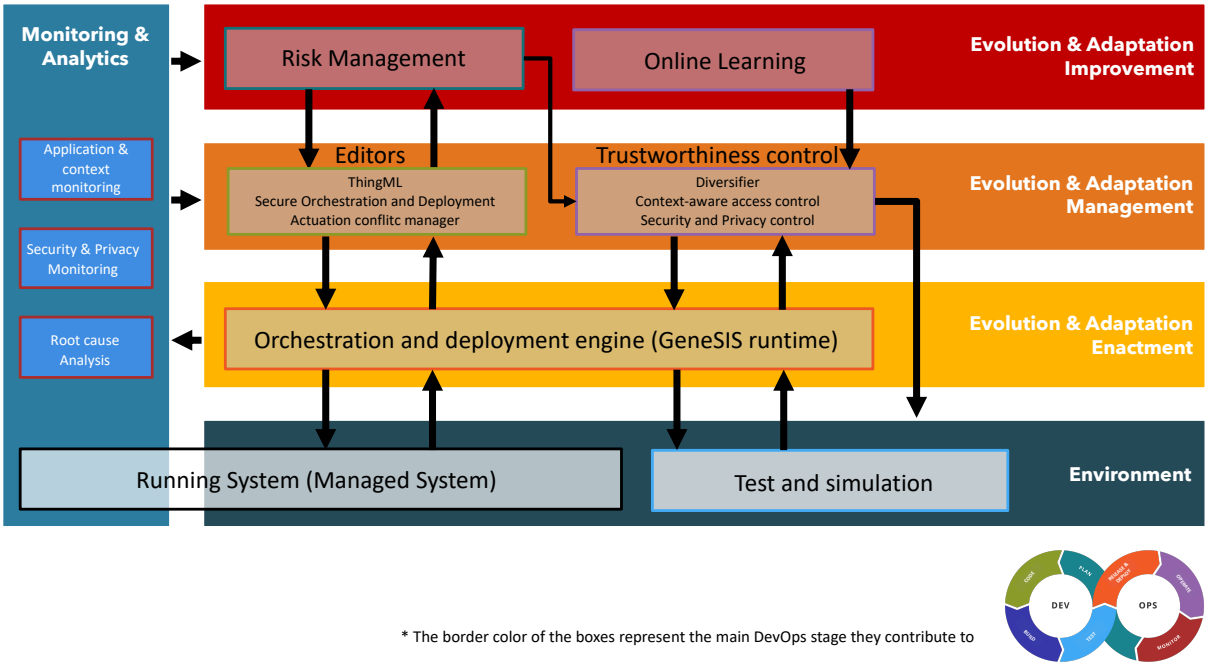


*Figure 1. The ENACT Multi-Layered Architecture*

## 2.1. Instruction of the framework

Within this section a showcase of the enablers is done, with a short description and a pointer on how to install and use the tool. All enablers except for Risk Management are hosted within open source Gitlab repository and available under https://gitlab.com/enact group. Each of the repository has a unified structure and is following the well accepted standards of open-source folder structure.

| Enabler: Risk management | Responsible partner: BEA | WP: WP2 |
|---|---|---|
| **Tool presentation / description:**<br>ENACT Risk Management opens the scope of the risk assessment to any type of risks where the user is free to express the scope of risks from non-intangible non-technical risks down to the tangible technical risks, which in effect dictate actionable mitigation actions that need to be included in the DevOps process. The novelty of the enabler comes from the following fact: Risk Management shall be approached in a continuous and agile fashion, which the tool facilitates. ENACT Risk Management enabler aims at embedding risk management in an agile development context in a non-intrusive way. In particular, the tool tries to solve several different challenges, namely:<br>• Traditional risk analysis practices for software development do not easily translate to Agile.<br>• Analysis of risks should be continuous.<br>• Development teams (i.e., scrum teams) do not have enough expertise on risk analysis.<br>• Tools to manage risk in Agile do not foster collaboration.<br><br>**Publications ENACT:** | | |

| |
|---|
| **"Enabling Continuous Privacy Risk Management in IoT Systems"** in Security Risk Management for the Internet of Things: Technologies and Techniques for IoT Security, Privacy and Data Protection. Edited by John Soldatos. pp. 143–160. Now Publishers. DOI: 10.1561/9781680836837.ch8.<br>**"Model-based Privacy-related Continuous Risk Control for Trustworthy Smart IoT Systems"**. Muntés-Mulero, V., Dominiak, J., González-Vidal, E., & Sanchez-Charles, D. (2019). In MDE4IoT/ModComp@ MoDELS (pp. 23-30), |
| **How to install:**<br>https://github.com/eclipse-researchlabs/enact-rm-API/blob/main/docs/INSTALL.md |
| **Tutorials:**<br>https://github.com/eclipse-researchlabs/enact-rm-API/blob/main/docs/GUIDE.md |
| **Videos / webinar:**<br>*Risk Management analysis example:*<br>https://www.youtube.com/watch?v=a0c1C8pc6sE |
| **License:** EPL-2.0 (due to close relationship of Beawre with the Eclipse foundation, the enabler is released under Eclipse license. The project will be available as a standalone Eclipse trademarked open source project) |

| Enabler: ThingML | Responsible partner: Tellu | WP: WP2 |
|---|---|---|
| **Tool presentation / description:**<br>ThingML is composed of *i*) a **modelling language**, *ii*) a set of **tools** and *iii*) a **methodology**. The modelling language combines well-proven software modelling constructs for the design and implementation of distributed reactive systems:<br>• statecharts and components (aligned with the UML) communicating through asynchronous message passing<br>• an imperative platform-independent action language<br>• specific constructs targeted at IoT applications.<br>The ThingML language is supported by a set of tools, which include editors, transformations (e.g. export to UML) and an advanced multi-platform code generation framework, which support multiple target programming languages (C, Java, Javascript). The methodology documents the development processes and tools used by both the IoT service developers and the platform experts. In the context of ENACT, ThingML has been extended with a platform independent mechanism for logging heterogeneous targets (from devices running in the cloud to tiny Arduinos).<br>**Publication in ENACT:**<br>**"Model-based, Platform-independent Logging for Heterogeneous Targets".** *Brice Morin, Nicolas Ferry* in the proceedings of the IEEE/ACM MODELS conference, Munich, Germany, September 15-20, 2019 | | |
| **How to install:**<br>https://github.com/TelluIoT/ThingML | | |
| **Tutorials:**<br>*Getting started tutorials:*<br>https://github.com/TelluIoT/ThingML/tree/master/doc/GettingStarted<br>*Tutorials for MODELS 2018:*<br>https://github.com/TelluIoT/ThingML/tree/master/doc/MODELS18-tutorial | | |
| **Videos / webinar:**<br>*SMOOL ThingML Integration:*<br>https://www.youtube.com/watch?v=mfT_AwfkXNc | | |
| **License:** MIT | | |

| Enabler: GeneSIS | Responsible partner: SINTEF | WP: WP2 |
|---|---|---|
| **Tool presentation / description:** | | |

GeneSIS supports the automatic deployment of software, together with the attached security mechanisms, across the computing continuum from IoT, Edge to Cloud. Developers use a declarative modelling language to specify what software components and security mechanisms they want to deploy, and the engine automatically deploys them into the resources in the computing continuum, and monitors the deployment status. The GeneSIS language includes security mechanisms as the first-class modelling elements. The engine implements a new concept of deployment delegation to support the devices with constrained resources or connectivity. It uses models at runtime to support declarative deployment with high availability and monitoring the deployment process.

**Publications in ENACT:**
**"GeneSIS: Continuous Orchestration and Deployment of Smart IoT Systems"** . *Nicolas Ferry, Phu H. Nguyen, Hui Song, Pierre-Emmanuel Novac, Stéphane Lavirotte, Jean-Yves Tigli, Arnor Solberg* Short paper in the proceedings of the IEEE COMPSAC conference, Milwaukee, USA, July 15-19, 2019

**"Continuous Deployment of Trustworthy Smart IoT Systems"** . *Nicolas Ferry, Phu Nguyen, Hui Song, Erkuden Rios, Eider Iturbe, Angel Rego Fernandez, Satur Martinez* in Journal of Object Technology (JOT), special issue for ECMFA, AITO, 2020

**"Towards Model-Based Continuous Deployment of Secure IoT Systems"** . *Nicolas Ferry, Phu H. Nguyen* in the proceedings of the DevOps@MODELS International Workshop, IEEE, Munich, Germany, September 2019

**How to install:**
https://enact.gitlab.io/GeneSIS/#/./install

**Tutorials:**
*Deploy Node-Red via Docker*
https://enact.gitlab.io/GeneSIS/#/./tutorial/1.nodered_localhost/

*Build and deploy ThingML programs*
https://enact.gitlab.io/GeneSIS/#/./tutorial/2.thingml_localhost/

*Deploy multiple instances of a same program*
https://enact.gitlab.io/GeneSIS/#/./tutorial/3.two_nodered/

*Deploy using Ansible*
https://enact.gitlab.io/GeneSIS/#/./tutorial/4.Ansible_resources/

*Deployment using regular SSH*
https://enact.gitlab.io/GeneSIS/#/./tutorial/5.SSH_resources/

*The deployment agent*
https://enact.gitlab.io/GeneSIS/#/./tutorial/6.Deployment_Agent/

*Blue/green deployment*
https://enact.gitlab.io/GeneSIS/#/./tutorial/7.blue_green_deployments/

**Videos / webinar:**
*Deployment of the ITS use case (2019):*
https://www.youtube.com/watch?v=rIoNcIAK3Kg

*Integration between GeneSIS and Jenkins:*
https://www.youtube.com/watch?v=RRFYirWiZ_0&t=8s

*Smart home use case demonstration:*
https://www.youtube.com/watch?v=RRFYirWiZ_0&t=8s

*DevOps IoT Deployment tool GeneSIS for Continuous Enhancement of Security Controls:*
https://www.youtube.com/watch?v=yQ9XYWu-EZM

*Continuous Deployment of Trustworthy Smart IoT Systems", Journal of Object Technology:*
https://youtu.be/if4DF9NPYPE

| **License:** Apache v2.0 |
|---|

| **Enabler: Actuation Conflict Management** | **Responsible partner: CNRS** | **WP: WP2** |
|---|---|---|

**Tool presentation / description:**
IoT applications have been limited to collecting field information for a long time. However, Smart IoT Systems (SIS) are now not only composed of sensors, but also of actuators. Therefore, in case of infrastructures with shared actuators, interactions with the physical environment raise additional challenges that cannot be ignored. Actuators not only raise concurrent and possibly conflicting access problems. They also raise problems of semantic coherency between the considered actions and their resulting effects in the environment (e.g., opening a window while heating). The consequences of actions and their impacts in the physical environment may put at risk applications functionalities. Indeed, applications being no longer isolated processes, they are not immune to the effects of the concurrent applications sharing the same environment and potentially producing antagonistic effects. While traditionally, the management of actuation conflicts is handled at the code level, the Actuation Conflict Management enabler applies over and abstract representation of the SIS that is decoupled from its detailed code enabling the detection, analysis and resolution of actuation conflicts as part of a typical DevOps process. DevOps team can integrate the enabler as part of their DevOps pipeline to detect automatically direct and indirect actuation conflicts in a complex smart IoT system. Off-the-shelf actuation conflict managers are automatically injected into the SIS. New actuation managers can be designed using a tool-supported domain-specific modelling language and checked against logical and temporal properties.

**ENACT publications:**
"**An Actuation Conflicts Management Flow For Smart IoT-based Systems**". Gérald Rocher, Thibaut Gonnin, Franck Dechavanne, Stéphane Lavirotte, Jean-Yves Tigli, Laurent Capocchi et Jean-François Santucci, in the proceedings of the 7th International Conference on Internet of Things: Systems, Management and Security (IOTSMS), December 2020. IEEE.

"**Actuation Conflict Management Enabler for DevOps in IoT**". Thibaut Gonnin, Franck Dechavanne, Gérald Rocher, Stéphane Lavirotte, Jean-Yves Tigli, Laurent Capocchi et Jean-François Santucci, in the proceedings of the IoT'20 Companion: 10th International Conference on the Internet of Things, October 2020. ACM.

**How to install:**
https://gitlab.com/enact/actuation_conflict_manager/-/blob/master/src/README.md#dependencies

**Tutorials:**
*Using ACM to solve direct conflicts in a Node-RED application*
https://gitlab.com/enact/actuation_conflict_manager/-/blob/master/docs/tutorials/1.node-red/README.md

*Using ACM to solve indirect conflicts in a Node-RED application*
https://gitlab.com/enact/actuation_conflict_manager/-/blob/master/docs/tutorials/2.node-red-env/README.md

*Using ACM with a GeneSIS deployment model*
https://gitlab.com/enact/actuation_conflict_manager/-/blob/master/docs/tutorials/3.genesis/README.md

*How to change the default configuration of ACM*
https://gitlab.com/enact/actuation_conflict_manager/-/tree/master/docs/tutorials/4.detection-configuration

**Videos / webinar:**
*ACM Smart Home use case demonstration*
https://www.youtube.com/watch?v=hxExx-eqEhk

*ACM ITS demos presentation*
https://www.youtube.com/watch?v=RXHkpAyxmt0

*ENACT Actuation Conflict Management (Release November 2019)*
https://www.youtube.com/watch?v=9S58MEgbk_s

*The ENACT Actuation Conflict Management Enabler*
https://www.youtube.com/watch?v=f1Fk7T14W10

**License:** Apache v2.0 (for the enabler), MIT (ACM nodes)

| **Enabler: Test and simulation** | **Responsible partner: MI** | **WP: WP2** |
|---|---|---|

**Tool presentation / description:**
Software testing is a crucial step of any software development process, even more in the DevOps software development cycle. However, in the development of IoT oriented applications, to have access to a production-like testing environment that reproduces the same conditions under which software would run is usually tricky and often an impossible task. Particularly, in IoT environments developers need to test their applications to ensure trustworthiness factors are met and guaranteed. Test and Simulation tool helps IoT Application developer save time, save money, and faster develop the IoT application. The main features of Test and Simulation:

- A Software as a Service solution
- Flexible simulation sensor model
- Powerful Data Generator
- Real-time data recording
- Support many testing types
- Support CI/CD

**How to install:**
Can use the docker image or install as a standalone application. More detail: https://gitlab.com/enact/test_and_simulation/-/wikis/home#installation

**Tutorials:**
Full tutorial can be found here:
https://gitlab.com/enact/test_and_simulation/-/wikis/home#tutorial

**Videos / webinar:**
*The Test and Simulation Enabler demonstration:*
https://www.youtube.com/watch?v=9h8q8f3oIHI

**License:** MIT

| Enabler: Online Learning | Responsible partner: UDE | WP: WP3 |
|---|---|---|

**Tool presentation / description:**
SIS are a special type of self-adaptive system (SAS) that need to continuously adapt themselves to changing context situation during runtime. Typically, adaptation rules define how such system adapt to react to changing context situation. As it infeasible to formulate such adaptation rules a priori, as not all context situations can be anticipated, a means to enable a SAS to learn such rules during run-time is desirable. The Online Learning Enabler (OLE) is one such means providing concepts to address the aforementioned problems. The corresponding tool can be used by a DevOps-engineer to apply Reinforcement Learning algorithms during run-time to an arbitrary SAS whose underlying adaptation process can be formulated as a sequential decision-making process. Being easy to configure and providing suitable means for monitoring the learning process, the OLE tool can be used to continuously explore the applicability and impact of using Online Reinforcement Learning techniques to tackle an identified adaptation process.

**ENACT publications:**
Palm, Alexander, Andreas Metzger, and Klaus Pohl. **"Online Reinforcement Learning for Self-adaptive Information Systems."** *International Conference on Advanced Information Systems Engineering*. Springer, Cham, 2020.
Metzger, Andreas, Tristan Kley, and Alexander Palm. **"Triggering Proactive Business Process Adaptations via Online Reinforcement Learning."** *International Conference on Business Process Management*. Springer, Cham, 2020.
Metzger, Andreas, et al. **"Feature Model-Guided Online Reinforcement Learning for Self-Adaptive Services."** *International Conference on Service-Oriented Computing*. Springer, Cham, 2020.
Palm, Alexander, et al. **"Towards Online Reinforcement Learning for Self-adaptive Fog Systems."** *KuVS-Fachgespräch Fog Computing 2020*: 8.

**How to install:**
https://gitlab.com/enact/online-learning-enabler/-/blob/master/readme.md#installation

**Tutorials:**
Exemplary setup of tool based on HVAC example:
https://gitlab.com/enact/online-learning-enabler/-/tree/master#exemplary-use-case
Endpoint description for integration of third party tools:
https://gitlab.com/enact/online-learning-enabler/-/tree/master#integrating-third-party-tools

**Videos / webinar:**

| |
|---|
| *Way of working (Mid-term-review version):*<br>https://www.youtube.com/watch?v=OD3iGChS8zw<br>Way of working (Final version):<br>https://youtu.be/-0OW3Jts15s |
| **License:** Apache v2.0 |

| **Enabler: Behavioural Drift Analysis** | **Responsible partner: CNRS** | **WP: WP3** |
|---|---|---|
| **Tool presentation / description:**<br>Smart IoT Systems (SIS) are computing systems composed with distributed computational elements whose, when embedded in physical things, a.k.a. devices, provide these systems with an interface to the physical world through transducers (sensors and actuators). These systems pose new challenges, as far as physical things are concerned, no guaranty can be made on their availability on the long run. The underlying infrastructure of SIS can thus be volatile. Moreover, the purpose of some of these systems can only be achieved from interactions with the physical environment through actuators (*e.g.*, Heating, Ventilation and Air-Conditioning (HVAC) controllers). In this context, these systems can possibly be affected by unanticipated physical processes over which they have no control leading their behaviour to potentially drift over time in the best case or malfunction in the worst case. Many platforms include context awareness and monitoring mechanisms. However, these platforms do not consider behavioural drift monitoring and analysis as an awareness criterion. Behavioural Drift Analysis (BDA) tool provides a way to monitor and analyse the behavioural drift symptoms of a SIS. In other words, it is a generic tool to detect when and how an application that has not changed is less effective in achieving its goals than expected. DevOps teams use BDA during operation as a monitoring solution to detect symptoms indicating that the effects of the system on its environment are no longer as expected and to understand this loss of effectiveness. BDA uses a new generic algorithm for learning behavioural models from the system operation, and a novel approach to identify the drift from the expected model to the actual one, using graph dissimilarity metrics.<br><br>**ENACT publications:**<br>Gérald Rocher, Stéphane Lavirotte, Jean-Yves Tigli, Guillaume Cotte et Franck Dechavanne. « **An IOHMM-based Framework to Investigate Drift in Effectiveness of IoT-based Systems ».** *in MDPI Sensors* **2021**, *21*(2), 527. | | |
| **How to install:**<br>https://gitlab.com/enact/behavioural_drift_analysis/-/blob/master/src/README.md | | |
| **Tutorials:**<br>*Defining and Deploying Behavioural Drift Computation model*<br>https://gitlab.com/enact/behavioural_drift_analysis/-/blob/master/docs/tutorials/1.bdc_model/README.md | | |
| **Videos / webinar:**<br>*Behavioural Drift Computation on a simulated train (ITS use-case)*<br>https://www.youtube.com/watch?v=mokA22r20a4 | | |
| **License:** Apache v2.0 (for the enabler) & Commercial (for legacy parts) | | |

| **Enabler: Root cause analysis** | **Responsible partner: MI** | **WP: WP3** |
|---|---|---|
| **Tool presentation / description:**<br>In complex systems, determining the causal factors of observed anomalies can be drastically difficult and time consuming due to the exceeding data sources (e.g., logs, traffic, metrics) identifying the status of the system. Root Cause Analysis's (RCA) objective is to infer the root-causes of problems by analysing the causal chains governing the system under monitoring. RCA plays a vital role in the Risk Management process (which normally includes vulnerability scanning, anomaly detection, root-cause analysis, and remediation). The tool is applicable to all systems in which collecting monitoring data is possible. In principle, it consists of the construction of a historical database of known/learned incidents (together with their corresponding symptoms, root-causes, impacts and mitigation actions based on the experts' experience), as well as the calculation of the similarity between the symptoms of new incidents with the ones stored in that database. Thus, it requires, enough relevant monitoring data attributes (for learning and for monitoring) and significant domain and system knowledge for the efficiency and accuracy of the analysis. | | |
| **How to install:**<br>*Dependencies:*<br>https://gitlab.com/enact/root_cause_analysis/-/blob/master/root_cause_analysis/package.json<br>*Installation:*<br>https://gitlab.com/enact/root_cause_analysis/-/tree/master/root_cause_analysis | | |
| **Tutorials:** | | |

| |
|---|
| https://gitlab.com/enact/root_cause_analysis/-/blob/master/root_cause_analysis/README.md |
| **Videos / webinar:**<br>*Playlist (will be kept updated with new videos):*<br>https://www.youtube.com/playlist?list=PL8Qwr001wPnY4lsCBzuBv0h-xilmTfg8z<br>*RCA-Principal idea (September-2020):*<br>https://www.youtube.com/watch?v=0Hixis8QXo0&list=PL8Qwr001wPnY4lsCBzuBv0h-xilmTfg8z&index=2<br>*RCA Dashboard (Beta version- October 2020):*<br>https://www.youtube.com/watch?v=Ri4EU66qb4A&list=PL8Qwr001wPnY4lsCBzuBv0h-xilmTfg8z&index=1 |
| **License:** MIT (for the enabler) & Commercial (for legacy parts) |

| **Enabler: Security and privacy monitoring and control** | **Responsible partner: Tecnalia** | **WP: WP4** |
|---|---|---|
| **Tool presentation / description:**<br>The Security and Privacy Monitoring tool allows the IoT system operator to continuously monitor the security and privacy status of the SIS at different layers. The tool will capture and analyse data from multiple and heterogeneous sources such as raw data from network, system and application layers, as well as events from security monitoring components such as Intrusion Detection System/Intrusion Prevention System (IDS/IPS). It provides intelligent analysis based on the combination of signature-based detection and artificial intelligence-based detection of incidents and anomalies. The holistic dashboard produces an informed situational awareness of the overall system and makes it easy for DevOp teams to understand security status of the SIS and promptly react accordingly. Moreover, the monitoring part is seamlessly integrated with the Control Manager which is able to manage the behaviour of the SMOOL IoT Platform. These control mechanisms at application layer rely on new developed SOFIA SMOOL capabilities in terms of secure communication monitoring and control agents' management.<br><br>**Publication in ENACT:**<br>Gallon, A., Rios, E., Iturbe, E., Song, H., & Ferry, N. (2020). **Making the Internet of Things More Reliable Thanks to Dynamic Access Control**. Security and Privacy in the Internet of Things: Challenges and Solutions, 27, 61.<br>Ferry, N., Nguyen, P. H., Song, H., Rios, E., Iturbe, E., Martinez, S., & Rego, A. (2020). **Continuous Deployment of Trustworthy Smart IoT Systems**. The Journal of Object Technology. | | |
| **How to install:**<br>*Control and monitoring in SMOOL IoT Platform:*<br>https://gitlab.com/enact/smool_enact | | |
| **Tutorials:**<br>*Usage of SMOOL clients for ENACT and ThingML integration:*<br>*https://gitlab.com/enact/smool_enact*<br>Security and privacy monitoring tool: The user's guide is available on demand. | | |
| **Videos / webinar:**<br>*The Control Manager at SMOOL IoT Platform:*<br>https://youtu.be/nnzTfVelcLE<br>*SMOOL IoT Platform Control with GeneSIS for Continuous Enhancement of Security Controls:*<br>https://www.youtube.com/watch?v=yQ9XYWu-EZM<br>*SMOOL ThingML Integration:*<br>https://www.youtube.com/watch?v=mfT_AwfkXNc | | |
| **License:** EPL (Eclipse Public License) (SMOOL IoT Platform extension) & Commercial (Security and privacy monitoring) | | |

| **Enabler: DivEnact** | **Responsible partner: SINTEF** | **WP: WP4** |
|---|---|---|
| **Tool presentation / description:**<br>DivEnact supports automatic software assignment and deployment of IoT applications to a large fleet of devices, and maintains software diversity among the fleet. IoT software vendors maintain multiple versions of software which are tailored to specific context properties of edge systems. Given that a fleet may comprise of hundreds and thousands of devices each with unique cyber-physical and user contexts, it is important to automate the assignment - i.e. what software components must be deployed on which devices. After each DevOps iteration, developers deploy the new software version into the abstract fleet, without worrying about what exact devices are in the fleet, their contexts, and whether they are online or not. The challenge is the assignment of multiple software variants to | | |

large number of devices according to their contexts. We address the challenge by converting it into the constraint solving and resource assignment problem and utilize automatic tools targeting these problems.

**ENACT Publications:**
**"Model-Based Automatic Fleet Deployment of Edge Computing Applications".** *Hui Song, Rustem Dautov, Nicolas Ferry, Arnor Solberg and Franck Fleurey* in the proceedings of the IEEE/ACM MODELS conference, 2020

**"A Light-Weight Approach to Software Assignment at the Edge"** *Rustem Dautov, Hui Song, Nicolas Ferry,* CloudAM'20, Leiceister, UK.

**How to install:**
*Full installation:*
https://gitlab.com/enact/divenact/-/blob/master/README.md#getting-started

*Simplified installation for a preview of the assignment approach, without Azure IoT Edge and real deployment of software on devices:*
https://gitlab.com/enact/divenact/-/blob/master/README.md#really-quick-start

**Tutorials:**
https://gitlab.com/enact/divenact/-/blob/master/README.md

**Videos / webinar:**
https://www.youtube.com/watch?v=KYNV0QB2njU

**License:** Apache v2.0

| Enabler: Context-Aware Access Control | Responsible partner: Evidian | WP: WP4 |
|---|---|---|

**Tool presentation / description:**

IoT systems link many devices such as sensors, cameras or smartphones to the Internet. These devices have the capacity to act as sensor or actuator in their environment, in a continuously changing context. Environmental data are considered as dynamic and give crucial information about a context (state of devices, user's behaviour, environment and location, etc.). The traditional mechanisms of access control do not use these contextual data while making authorization decisions.

The Context-Aware Access Control tool is a solution for context-based dynamic authorization, which allows to control in the same way the access of all the IoT actors (end-users, services, devices, administrators) to the operated data and resources, for both IT and OT (operational technologies) domains. In particular, this tool provides Context-aware risk & trust-based dynamic authorization mechanisms ensuring *(i)* that an authenticated IoT node accesses only what it is authorized to and *(ii)* that an IoT node can only be accessed by authorized software components. Access authorizations are adapted according to a risk level computed from contextual data on the user and his devices. This tool can be used in all IoT use cases where access control to protected resources is required. It is of special interest if this access control must consider the context and the resulting risk.

**ENACT Publications:**

Gallon, A., Rios, E., Iturbe, E., Song, H., & Ferry, N. (2020). **Making the Internet of Things More Reliable Thanks to Dynamic Access Control**. Security and Privacy in the Internet of Things: Challenges and Solutions, 27, 61.

**How to install:**
EVIDIAN Proprietary software provided in SaaS mode.

**Tutorials:**
EVIDIAN Proprietary software. The user's guide is available on demand.

**Videos / webinar:**
https://youtu.be/JKAYQqx1ShU

**License:**
Commercial

Within next section, we will go over the evolution of the user stores defined at the beginning of the project. User stores progression can give an understanding of the fulfilment of the initial requirements as well as if new requirements have been detected.

## 2.2. Final Enablers User Stories

### 2.2.1. Introduction

To facilitate the reading of the user stories status, we propose the following visual information:

- An achieved user story (as it was defined in D5.1 and D5.2).
- A modified user story (compared to the one specified in D5.1 and D5.2). The original user story is referenced in footnote.
- An unsuccessful user story
- A new added user story

### 2.2.2. Evolution & Adaptation Improvement Layer

#### *2.2.2.1. Online Learning Enabler*

| **Improve the behaviour of the system during operation.** |
|---|
| **Related User Stories:**<br>• As an IoT DevOps engineer, I want the online learning enabler to perform automatically, so that no human intervention is needed.<br>• As an IoT DevOps engineer, I want the online learning enabler to continuously self-improve the system's adaptation policies, so that the way the system adapts is improved.<br>**Description:**<br>This is the main feature of the OLE. It should improve a system during operation time by especially exploiting all other features listed. |
| **Enable the user to supervise/monitor the learning process.** |
| **Related User Stories:**<br>• As an IoT DevOps engineer, I want to observe the learning progress of the online learning enabler to evaluate the convergence of the learning behaviour.<br>• As an IoT DevOps engineer, I want the online learning enabler to converge fast, so that situations where my system is performing non-optimally are avoided.<br>**Description:**<br>Possibilities to monitor the learning progress increases the trustworthiness of user into the enabler. Through monitoring divergence in the learning behaviour could be identified so that the OLE might be reconfigured. Further means for pretraining the model used by the OLE (and targeting the 2nd user story of this paragraph) have been described in D3.3. |
| **Ease manual steps during design-time.** |
| **Related User Stories:**<br>• As an IoT DevOps engineer I want my system to be able to learn online, so that it can automatically resolve the uncertainties that I have as DevOps-engineer about its environment at design time.<br>• As an IoT DevOps engineer, I want the online learning enabler to be able to differentiate between different environment situations, so that it can cope with changing (i.e., non-stationarity) environment properties and behaviour.<br>• As an IoT DevOps engineer, I want the online learning to be able to deal with state and action spaces of arbitrary size, so that scalability is ensured. |

**Description:**
This feature sums up the main motivation behind online learning in general. By enabling a system to adapt itself during operation time, manual steps need during design-time can be accelerated or even be skipped.

**Efficient usage of experience for policy updates.**

**Related User Stories:**

- As an IoT DevOps engineer, I want the online learning enabler to be capable of learning even from small chunks of experience, so that the adaptation policies may be updated as frequently as possible.

**Description:**
Algorithm used to realize the behaviour of the OLE should be tuned to as sample-efficient as possible. Depending on Hyperparameters the set of samples needed for an update can be adjusted.

**Provide different ways to configure the corresponding tool.**

Related user stories:

- As an IoT DevOps engineer I want to configure the OLE tool via a Web API to have a proper interface.
- As an IoT DevOps engineer I want to configure the OLE tool via a Graphical User Interface to ease the configuration process.

**Description:**
This feature results from different user needs. Some users may want to integrate the tool into a proprietary user interface and directly configure the tool via RestAPI; others may quickly want to test the tool. In both cases, a proper documentation of the interface is inevitable.

**Online Learning tool isolated from system to be improved.**

**Related user stories:**

- As an IoT DevOps engineer I want the OLE tool to be run in its own environment and to be encapsulated, so that it can easily deployed on different locations.
- As an IoT DevOps engineer, I want the online learning enabler to produce as less overhead as possible, so that it can be used even on devices with low computing power.

**Description:**
With the corresponding OLE tool being isolated from the system that should be approved it is easier to integrate it. As it is running in a separate Docker container it can be deployed on its own device, so that it does not negatively influence the system to adapt through performance reasons. On the other hand, this offers the opportunity to deploy it as a cloud service.

**Handle changes during development time (e.g. introduction of new features).**

**Related user stories:**

- As an IoT DevOps engineer, I want my system to automatically take changes made at the development cycle (e.g. new features introduced during evolution) into consideration, so that the system performs optimally.

**Description:**
This Feature should enable the tool to seamlessly integrate into the DevOps-Cycle and take changes during design-time into consideration at operation time.

#### *2.2.2.2. Risk Management*

**Threats Kanban board**

**Related User Stories:**

- As an IoT application developer, I want to know what are my functional risks so I can ensure application trustworthiness
- As a Risk Manager, I want to be aware of the status of the detected functional and non-functional risks so I can manage my risks effectively.
- As a Risk Manager, I want to collaborate with other actors involved within the software development process, so I can ensure that all types of risks are owned.

**Description:**
This feature enables to monitor the status and the progression of the threats mitigation process, enables easy to consume, familiar view for the IoT application developers not only to assess the status but also to participate and influence the risk mitigation process.

**Treatment Dashboard**

**Related User Stories:**

- As a Risk Manager, I want to collaborate with other actors involved within the software development process, so I can ensure that all types of risks are resolved.
- As a Risk Manager, I want to collaborate with other actors involved within the software development process, so I can ensure that all types of risks are mitigated.
- As a Risk Manager, I want to collaborate with other actors involved within the software development process, so I can ensure that all types of risks are accepted.
- As a Risk Manager, I want to collaborate with other actors involved within the software development process, so I can ensure that all types of risks are accepted.
- As an IoT application developer, I want to know which functional risks are of the highest priority and what are the mitigation actions for them so I can plan the development including these actions.
- As a Risk Manager I want to be able to detect regressions of risks detected in an automatic way so I can assess if the mitigation actions are sufficient.

**Description:**
This feature enables to monitor the status and the progression of the threats mitigation process, enables easy to consume, familiar view for the IoT application developers not only to assess the status but also to participate and influence the risk mitigation process. This feature also enables the feedback from the collected evidences in order to ensure that the mitigations defined has been taken on board, which in effect closes the loop of the continues risk mitigation process.

**Evidence Collectors**

**Related User Stories:**

- As an IoT application developer, I want to know which functional risks are of the highest priority and what are the mitigation actions for them so I can plan the development including these actions.
- As an IoT application developer, I want to gain insights into the all types of risks so I can influence the technical requirements of the software.
- As a IoT application developer I want to integrate the risks into my agile process so that I naturally integrate it into my everyday work.
- As a Risk Manager I want to ensure that detected risks are not affecting the release schedule so that I can execute the releases as planned.

**Description:**
This feature ensures that the loop of the risk assessment is followed and that the process becomes the risk management process. The state of the evidence collectors provides the factual evidences of the actions taken in order to mitigate detected risks and ensures that the current understanding of the risk status is as accurate as possible. In the scope of the project, we do integrate with two major players of the DevOps cycle, that is Ticket Management and Project Planning software (Jira) and the code repository (git).

| Risk Assessment |
|---|
| **Related User Stories:**<br>• As a Risk Manager I want to be able reliably specify the risk likelihood and consequence based on agreed methodology.<br>• As a Risk Manager I want to be able to choose risk management actions which are best suited for the types of risks vs the architecture where the risk occurs so that the risk management is optimized for the scenarios faced.<br>• As a Risk Manager I want to be able to know what is the minimum viable risk levels where the "just enough" scenario of risk management is reached so I can be aware what mitigation actions are strictly necessarily.<br>**Description:**<br>This feature encapsulates the best practices of the risk assessment process in the scope of the specification of the unwanted incidents, vulnerabilities, assessment of the likelihood and impact and specification of the mitigation actions. The result is easy to consume and use feature which helps the user define the specification of risk based on the common standards used for the DevOps IoT space and beyond. |
| **Components Architecture View / DFD view** |
| **Related User Stories:**<br>• As a Solution architect I want to detect weak points of the architecture and the associated risks so that I can detect related risks.<br>**Description:**<br>This feature enables the audit of the status of the analysed subject, would that be architectural components of the application or Data Flow Diagram. Based on this feature, the assessment can be made on the most critical actions to be taken first, the cost and the impact on the current DevOps process. |

### 2.2.3. Evolution & Adaptation Management Layer

#### *2.2.3.1. ThingML*

| Monitor and debug the execution flow of a ThingML program. |
|---|
| **Related User Stories:**<br>• As an IoT DevOps engineer, I want to monitor the execution of my program in a platform independent way, so that I can debug it independently of the deployment target<br>• As an IoT DevOps engineer, I want to specify how logging messages are formatted (string or binaries) and transmitted over MQTT(s), so that I can best exploit it in my monitoring toolchain<br>**Description:**<br>ThingML has been extended with a logging mechanism. ThingML programs can be annotated with a monitoring instruction. The mechanisms to remotely monitor the annotated parts of the program automatically as ThingML code, meaning they can, in turn, be automatically deployed on heterogeneous devices. Logged messages can be formatted as String or binaries. |

#### *2.2.3.2. Secure Orchestration and Deployment (aka. GeneSIS)*

| Support the specification of Deployment model in a declarative way. (Done) |
|---|
| **Related User Stories:**<br>• As an IoT DevOps engineer, I want to specify the deployment of my SIS, so that I can in turn automatically deploy it.<br>• As an IoT DevOps engineer, I want to see the list of sensors and actuators involved in my SIS, so that I can manage and optimize their usage. |

- As an IoT DevOps engineer, I want to know the type of resources on which a SIS can be deployed, so that I can optimize my deployment.

**Description:**
This is the main design-time feature of GeneSIS. GeneSIS provides a domain-specific language with textual or graphical syntax to support the specification of the deployment of SIS in a declarative way. This includes the feature listed below.

**Support the specification of where and how to deploy software components**

**Related User Stories:**

- As an IoT DevOps engineer, I want to specify the deployment of my SIS, so that I can define how to orchestrate the deployable software components.
- As an IoT DevOps engineer, I want to specify the deployment of my SIS, so that I can decide where to deploy my system.

**Description:**
By using the concept of `Containment` in the GeneSIS modelling language it is possible to specify where component should be deployed. By using the concept of `Resources`, it is possible to specify how to manage the life-cycle of a component (start, stop, configure, etc.).

**Support the specification of security and privacy requirements in deployment models**

**Related User Stories:**

- As an IoT DevOps engineer, I want to specify the deployment of my SIS, so that I can define security mechanisms to be deployed as part of the system.

**Description:**
By attaching security and privacy capabilities to ports it is possible to specify the security and privacy features offered and required by the software components. Specific components offering security features can be added to the deployment to satisfy security requirements from other components.

**Check the validity of a deployment model**

**Related User Stories:**

- As an IoT DevOps engineer, I want to specify the deployment of my SIS, so that I can ensure the correctness of my future deployment.
- As an IoT DevOps engineer, I want to specify the relationship between the deployable software components, so that I can check the dependencies.
- As an IoT DevOps engineer, I want to specify the relationship between the software components to be deployed and actuators, so that I can identify concurrent accesses to actuators.
- As an IoT DevOps engineer I want to check that all the security requirements expressed I my deployment model are met, so that I can check that my deployment model is conform to my expectations

**Description:**
GeneSIS embeds a validation engine that check the correctness of a deployment model before its deployment, this include checking: (i) that all containments and communications have a source and a destination, (ii) that all the requirements in term of capabilities (including security and privacy) are fulfilled, (iii) a set of properties on the name, id and the different attributes of each entity in the model.

**Support the creation of new type of GeneSIS component**

**Related user stories:**

- As an IoT DevOps engineer I want to be able to create and save new GeneSIS component type so that I can reuse them afterwards

- As an IoT DevOps engineer, I want to see and access a set of off-the-shelf deployment components, so that I can use them in a one-click approach

**Description:**
New type can easily be created so they can later be instantiated in a deployment model. New types are dynamically loaded in the GeneSIS editor.

**Provide Specific GeneSIS components for the deployment of security and privacy mechanisms**

**Related user stories:**

- As an IoT DevOps engineer I want to be able to select an existing component so that I can deploy security and privacy mechanisms

**Description:**
DevOps engineers are provided with predefined components for security and privacy mechanisms (e.g., jCasbin, ExpressAPI).

**Support the automatic deployment of SIS over Cloud, Edge, and IoT resources.**

**Related user stories:**

- As an IoT DevOps engineer, I want to automatically deploy a SIS, so that I can automate the delivery of my system in a production environment.
- As an IoT DevOps engineer, I want to automatically deploy a SIS, so that I can reproduce the delivery of my system in a production environment.

**Description:**
This is the main feature of the execution engine of GeneSIS. It supports the automatic deployment of SIS from a deployment model written using GeneSIS. A deployment model can be replicated anytime from a same deployment model. This includes the feature listed below. (see D2.3)

**Support automatic deployment of blackbox components over Cloud, Edge, and IoT resources**

**Related user stories:**

- As an IoT DevOps engineer, I want to automatically deploy a SIS, so that I can automate the delivery on cloud, edge, and IoT devices of software components provided as binaries, scripts, etc. and available off the shelf as blackboxes.

**Description:**
GeneSIS support the deployment of SIS over Cloud, Edge, and IoT resources. The deployable artefacts (implementation of the components to be deployed) can be binaries or scripts available off-the-shelf for the deployment of generic or already existing solutions (e.g., servlet containers, application servers, libraries).

**Support automatic deployment of ThingML programs over Cloud, Edge, IoT resources**

**Related user stories:**

- As an IoT DevOps engineer, I want to automatically deploy ThingML programs, so that I can automatically deploy it on different cloud, edge, and IoT resources.

**Description:**
GeneSIS offers specific support for the deployment of ThingML programs. Depending on the targeted host of a ThingML program, GeneSIS automatically (i) compiles it toward the proper programming language, (ii) builds it, and (iii) deploys it. A program written in a platform independent way using ThingML can thus be automatically migrated from one host to another.

**Support Automatic deployment over devices with limited access to Internet**

**Related user stories:**

- As an IoT DevOps engineer, I want to automatically deploy a SIS, so that I can automate the delivery software components on resources with limited access to internet.

**Description:**
By using the concept of deployment agent, GeneSIS supports the deployment of software components on devices with limited access to Internet. In short, GeneSIS automatically generates the agent, deploys it on a device that has access to internet and to the device without access to internet and delegate to it the responsibility of deploying the software component.

**Support the deployment of security and privacy mechanisms**

**Related user stories:**

- As an IoT DevOps engineer, I want to automatically deploy a SIS, so that I can automate the deployment ENACT security and privacy mechanisms as part of my SIS.
- As an IoT DevOps engineer, I want specific ENACT security and privacy component types in GeneSIS, so that I can deploy them as part of my SIS by selecting them.
- As an IoT DevOps engineer, I want to specify security policies to be injected or to be reconfigured into the security mechanisms of my SIS, so that I can refine and extend them in a DevSecOps way.
- As an IoT DevOps engineer, I want to seamlessly integrate security mechanisms into IoT middleware, so that I can extend middleware security with third party solutions

**Description:**
As other software components, GeneSIS support the deployment of security and privacy mechanisms. Specific component types are available for the ENACT security and privacy mechanisms, thus facilitating their deployment. In particular, from a policy, the security control to be enacted within IoT platforms mechanism can be generated from ThingML. It is possible to configure from GeneSIS the security checkers in IoT middleware (SMOOL) or to inject these security checkers code into the middleware in a DevSecOps fashion. This injection enables the seamless integration of third party security solutions in IoT middleware.

**Continuously reflect the status of a deployment in a deployment model**

**Related user stories:**

- As an IoT DevOps engineer, I want to monitor the status of an ongoing deployment, so that I can check its completeness and functioning.
- As an IoT DevOps engineer, I want to monitor the execution of my IoT program (the program deployed on tiny devices), so that I can debug it.
- As an IoT DevOps engineer, I want to deploy a monitoring agent, so that I can observe the performances of my systems

**Description:**
The GeneSIS execution engine continuously reflect the status of the deployment in the deployment model. This is achieved by implementing the models@runtime pattern. Specific monitoring agent can be deployed in order to gather details about the performance of the system (CPU, RAM usage, etc.). In addition, ThingML programs can be monitored thanks to the new logging mechanisms, thus providing details about the execution of an IoT program.

**Support the continuous update/adaptation in the deployment of a SIS**

**Related user stories:**

- As an IoT DevOps engineer, I want to migrate part (or all) of my SIS from an infrastructure to another, so that I can ensure the trustworthiness of my SIS.
- As an IoT DevOps engineer, I want to update all or part of my SIS, so that I can improve its trustworthiness.

**Description:**
GeneSIS supports the dynamic adaptation of the deployment of a SIS. Software components can be updated and part or all the deployment can be modified, including the orchestration of the components. An adaptation can be performed in an imperative way – i.e., changes can be triggered one by one in a specific order; or in a declarative way – i.e., a new deployment model can be provided to the execution engine which in turn computes the necessary adaptation to move from the current system to the target one with minimal impact on the running system. In addition, blue/green deployment techniques are provided for the Cloud/Edge space.

**Support automatic encrypted communication between deployed components without any change in the component code**

**Related user stories:**

- As an IoT DevOps engineer, I want to secure communication between components of my SIS, so that I do not need to change their implementation

**Description:**
GeneSIS provides a mechanism to generate proxies between the communicating components, which are responsible for encrypting the communications.

**Support the automatic blue/green deployment of Cloud and Edge resources**

**Related user stories:**

- As an IoT DevOps engineer, I want to deploy my application using the blue/green deployment strategy, so that I ensure continuity of service
- As an IoT DevOps engineer, I want to specify my blue/green deployment strategy in a platform-independent way, so that I can reuse it across different targets
- As an IoT DevOps engineer, I want to select whether I delegate my blue/green deployment strategy to an existing platform or to GeneSIS, so that I can apply it to any type of software component.

**Description:**
DevOps engineer can seamlessly apply the blue/green deployment strategy (I.e., new components are deployed in parallel of existing ones, and replace them only if working properly). Thanks to our model-based approach, blue/green strategy can be specified in a platform-independent way. For container-based components, the deployment strategy can be applied leveraging the container platform solution. For other components, GeneSIS takes care of deploying and maintaining the blue/green deployment supporting services (i.e., load balancer with healthcheck) as well as of wrapping the software into containers.

#### *2.2.3.3. Actuation Conflict Management*

**Detect actuation conflicts**

**Related User Stories:**

- As an IoT DevOps engineer, I want the actuation conflict manager to detect actuation conflicts and show them to me, so that I can visualize where are the conflicts.
- As an IoT DevOps engineer, I want to customize the actuation conflict detection mechanism to add or modify conflict patterns.

**Description:**
The Actuation Conflict Management enabler helps detecting actuation conflicts present in a SIS to facilitate their solving. The Actuation Conflict Management enabler offers the developer the possibility to detect conflicts on a representation based on application deployment models. The detection is done by analysing the application models and using a physical environment model to define how actuators interact and lead to direct and indirect conflicts. The detection of actuation conflicts is based on a set of predefined rules. If desired or necessary these rules can be modified, removed, or new rules can be added. A specific language is provided for specifying the rules and how they can be applied.

| **Help developer to find the right actuation conflict manager** |
|---|
| **Related User Stories:**<br>• As an IoT DevOps engineer, I want to browse a palette of pre-made actuation conflict managers, so that I am able to know the range of solutions I have.<br>• As an IoT DevOps engineer, I want to visualize the code template used by an actuation conflict management policy inside the palette, so that I can determine whether it's suitable for my problem.<br>• As an IoT DevOps engineer, I want to select a list of keywords (and annotations) to parametrize which subset of conflict management policies is shown in the palette, so that I am able to select more efficiently the appropriate one.<br>**Description:**<br>The palette is composed of off-the-shelf actuation conflict managers implementing straightforward conflict management strategies. Each component available in the palette is complemented by metadata describing its properties and attributes. Such metadata can be then used to allow filtering of the components in the palette to ease choosing the adequate strategy or be used by the resolution mechanism to apply the policy adapted to the conflict resolution. |
| **Resolve actuation conflicts with off-the-shelf actuation conflict manager** |
| **Related User Stories:**<br>• As an IoT DevOps engineer, I want to be able to resolve actuation conflicts using off-the-shelf conflict management policies in my application, so that I can manage inputs to the actuators used in application and reduce misbehaviour.<br>• As an IoT DevOps engineer, I want to add actuation conflict manager to my application assembly, so that I can manage inputs to the actuators to reduce misbehaviour.<br>• As an IoT DevOps engineer, I want to parametrize the conflict management policy I chosen to resolve a specific conflict, so that I can set parameters to suit that conflict.[1]<br>• AS an IoT DevOps engineer, I want to visualize the custom actuation conflict manager DSL, so that I can determinate whether it's suitable for my problem.[2]<br>**Description:**<br>Actuation Conflict Management is providing a set of default off-the-shelf actuation managers to resolve detected conflicts. Two user stories have been modified as our solution went beyond expectation in term of user support. It is no longer necessary to configure the actuation conflict manager inputs and outputs, as it is individually generated for each case. The FSM visualization is generated from the ECA+ language and would not help the developer (in some case, it is as if we provide the assembly code to a C developer, it is not really helpful to find an issue). |
| **Enhance policies to identify and resolve conflicts** |
| **Related User Stories:**<br>• As an IoT DevOps engineer, I want to be able to develop my own conflict management policy, so that I can implement a new behaviour not present in the off-the-shelf palette.<br>• As an IoT DevOps engineer, I want to write code defining a new conflict management policy template from a code skeleton and the same inputs and outputs as off-the-shelf conflict management components, so that I am able to produce a custom policy that can be used as drop-in replacement for an off-the-shelf node. |

[1] The original user story was: As an IoT DevOps engineer, I want to parametrize the conflict management policy I chosen to resolve a specific conflict, so that I can set the number of inputs, outputs and other parameters to suit that conflict.
[2] The original user story was: As an IoT DevOps engineer, I want to visualize the FSM used by an actuation conflict manager, if it relies on an FSM, inside the palette, so that I can determine whether it's suitable for my problem.

- As an IoT DevOps engineer, I want my new conflict management policy to be available in the palette among other off-the-shelf conflict management policies, so that it can be selected and added to the application to solve a conflict.
- As an IoT DevOps engineer, I want to insert keywords and annotations describing my custom policy, so that it can be available in the filtered palette.

**Description:**
Actuation Conflict Management is based on a generic approach using rules (see D2.3) to identify and resolve actuation conflicts. The objective is to provide a generic and extensible approach to modify, add, remove policies (model transformation rules) to adapt the actuation conflict detection and resolution.

**Ensure logical and temporal behaviour when designing new actuation conflict managers**

**Related User Stories:**

- As an IoT DevOps engineer, I want to be able to develop a new conflict management policy, so that I can implement a new behaviour, not present in the off-the-shelf palette, with guaranteed logical and temporal behaviour.
- As an IoT DevOps engineer, I want to design a new actuation conflict manager with a domain specific language (DSL) to specify a safe and custom actuation manager defining and verifying logical and temporal behaviour.[3]
- As an IoT DevOps engineer, I want to be able to select the event generation strategy using the DSL.[4]
- As an IoT DevOps engineer, I want my policy to be checked to ensure that it will function as intended, so that I can validate the behaviour before using the policy.
- As an IoT DevOps engineer, I want my design to be translated into a code template for an actuation conflict manager available in the palette to use it in my assembly, so that I can use that new conflict management component in an application.

**Description:**
The behaviour of the actuation conflict manager is specified using a domain-specific language. Once the actuation conflict manager logical and temporal behaviour has been defined, it is necessary to verify the expected logical (using NuSMV, a symbolic model checker) and temporal (using DEVS, Discrete Event System Specification) behaviour with model checkers before generating actuation conflict manager code. Two user stories have been modified as our solution went beyond expectation in term of user support. Instead of presenting the information to the programmer with a FSM representation, we have defined a domain specific language that is transformed into its FSM equivalent.

#### 2.2.3.4. DivEnact

**Appoint a universal deployment for all devices**

**Related User Stories:**

- As an IoT DevOps Engineer, I want to design a single generic deployment configuration and universally apply it to the overall fleet of devices.

**Description:**
All devices will be upgraded automatically when this universal deployment is updated. When a new device is bootstrapped, or an existing device is reconnected, they will be updated to this universal deployment. This is the baseline feature of DivEnact. The only diversity comes from the fact that we allow the devices to be upgraded in an asynchronous way: the ones that are not currently online still use the old version, until

[3] The original user story was: As an IoT DevOps engineer, I want to design the finite state machine modelling the new actuation conflict management policy, so that it is possible to formally define the behaviour of the new policy.
[4] The original user story was: As an IoT DevOps engineer, I want to be able to select the event generation strategy for my FSM-based custom policy, so that it can be deployed and manage inputs to the model.

they go online again. This feature is implemented by creating a production deployment model, setting its target condition to be “all the production devices”, and tagging all the managed devices as “production”.

**Appoint a subset of devices for preview**

**Related User Stories:**

- As an IoT DevOps Engineer, I want to design a deployment configuration and apply it only to specific devices in the fleet to test new functionality in a preview environment.

**Description:**
The devices in the preview environment will be deployed with the “next version” of the application, so that the vendor can collect the user feedbacks before pushing it to production (i.e. to all other users). The operator can select trial devices manually or specify a target condition to let the tool automatically select a specific sub-set of devices. This feature is implemented by creating a deployment model with the preview versions of the application modules and tagging the appointed or selected devices with the “preview” tag. The operator can also opt to keep an appointed number of preview devices, which means that if a preview device is offline, a next available one will be switched to the preview mode.

**Promote preview deployment to production**

**Related User Stories:**

- As an IoT DevOps Engineer, I want to apply an already tested and working deployment configuration currently running in a preview environment on a sub-set of devices to the overall fleet in production.
- As an IoT DevOps Engineer, I want to deploy the preview configurations on a particular number (such as 10% of all devices) of randomly selected devices.

**Description:**
All the devices will be upgraded to the preview deployment. This is implemented by updating the target condition of the preview deployment, so that it applies to all the production devices, and tagging the preview devices back to “production”.

**Diversify the applications based on a hardware condition**

**Related User Stories:**

- As an IoT DevOps Engineer, when designing a deployment configuration, I want to specify hardware capabilities of target Edge devices.

**Description:**
DivEnact maintains a set of equivalent candidate deployments, each of which applies to a specific setup of an edge device, including its CPU architecture, available resources, and IoT devices connected to it. This feature is implemented by adding another dimension of tags to the edge devices and control the conditions of the deployments to include these tags.

**Diversify the applications among identical devices**

**Related User Stories:**

- As an IoT DevOps Engineer, I want to distribute the candidate software variants among all the devices, to achieve software diversity among the entire fleet.[5]

**Description:**
The design-time diversifier can generate multiple versions of modules and architectures that are functionally equivalent. For the sake of security, operators can use DivEnact to shuffle these equivalent applications among the identical edge devices. We achieve this feature by generating an additional dimension of tags for the devices, and control the conditions of the deployment models, accordingly. We

[5]The original user story was: As an IoT DevOps Engineer, I want to apply "synthetic" diversity to the fleet of managed devices to increase system security and resilience without affecting its functionality.

extend the scope from only synthesized software variants to generic software variants, including the ones developed specifically for hardware capacities. In this way, the diversification among the entire fleet is more generic.

**Recover failed device**

**Related User Stories:**

- As an IoT DevOps Engineer, I want to ensure that a failed device after a factory reset can be bootstrapped with a default deployment configuration and corresponding software modules via simple manual instructions
- As a Device owner, I want to be able to manually reset the device to its factory settings.

**Description:**
If an edge device has software failures, the user can reset the device to the factory settings. Once it again gets connected back to the Internet, the deployment manager will automatically apply a deployment with respect to the current target conditions and push software modules according to the newest relevant deployment model. Due to hardware complexity, we did not meet the previous objective of having the devices to fetch all the relevant software modules by themselves. Instead, we require the system operators to remotely execute a command for bootstrapping.

**Rollback devices**

**Related User Stories:**

- As an IoT DevOps Engineer, I want to be able to recover an Edge device to a previous working deployment configuration.

**Description:**
DivEnact keeps track of applied deployment configurations and stores them in a repository. If a deployment has problem on a device, the operator can remotely roll it back to the last working deployment, which is recorded as a specific deployment for this device.

#### 2.2.3.5. Context-Aware Access Control

**Control the access of all the actors (end-users, services, devices, administrators) to the operated data and resources, for both IT and OT (operational technologies) domains.**

**Related User Stories:**

- As a Device owner, I want to be sure that only authorized people can access my personal data handled by my device to avoid leaks of my privacy data.
- As an Administrator, I want to be sure that only authorized people can consult the data gathered by the device to avoid leaks of the privacy data.

**Description:**
This is the main feature of the Context-Aware Access Control. It ensures confidentiality in the data managed by SIS, since information is not made available or disclosed to unauthorized individuals, entities, or processes. It should provide security and privacy to operation side by especially exploiting all the features listed below (see D4.3).

**Ensure privacy in the data managed by SIS, since the control over the personal data is kept by their owner.**

**Related User Stories:**

- As a Device owner, I want to be aware of my personal data handled by my device so that I remain the owner of my personal data.
- As a Device owner, I want to give my consent about the scope of personal data handled by my device so that I can check and eventually reduce this scope to be sure that the device handles only data I agreed.

- As a Device owner, I want to be able to consult all the data gathered by my device at any time so that I can get all my privacy information.

**Description:**
During the device enrolment phase, the user (owner of the device) accepts or declines the scopes requested by the device. The Authorization server then establishes and stores a link between the device and the user. At any time, the user can revoke the consented scopes.

**Ensure efficiency in the data managed by SIS, since information is made available or disclosed to authorized individuals, entities, or processes, in a controlled way.**

**Related User Stories:**

- As a Device user, I want to consult the data gathered by the device so that I can use them to perform the tasks I am responsible for.
- As an Administrator, I want to be sure that all authorized people can consult the data gathered by the device to allow them performing the tasks they are responsible for.

**Description:**
At any time, the authorized users can access the data produced by the connected objects in a controlled way.

**Access control adaptation depending on contextual data on the user and his devices.**

**Related User Stories:**

- As a Device user, I want to consult extra data in case of emergency to be able to react even by overriding my usual access rights.
- As an Administrator, I want to define contextual risk levels that widen the scope of data that authorized people can consult to be able to define how the privacy data can be handled in case of emergency.[6]

**Description:**
Adapt the provided authorizations according to a risk level computed from the contextual information. This risk value is faced to the authorization policy to define the granted authorizations. The authorization policy is a set of rules that define whether a user or device must be permitted or denied accessing a resource depending on risk values. Unlike what was originally planned, the provided authorizations are adapted according to a risk level associated to each user, instead of defining contextual events to be considered. Indeed, this modification allows to make the mechanism more generic (through the notion of risk level) while it can be completely customizable (since the risk level is computed from contextual information).

#### 2.2.3.6. Security and Privacy Control

**Thing trustworthiness control by SMOOL SOFIA IoT platform**

**Related User Stories:**

- As a system operator, I want to apply at runtime security controls in the IoT system.
- As a system operator, I want to be able to allow or deny communications in case symptoms are detected about the smart thing not being trustworthy or legitimate entity.

**Description:**
This enabler can configure a set of rules to detect application or network potential threats and react to them. The enabler has insights of system communication metadata and according to them it can decide to perform automatic actions like blocking transmissions to and from untrustworthy devices and sending alarms to the monitoring tool.

[6]The original user story was: As an Administrator, I want to define contextual events that widen the scope of data that authorized people can consult to be able to define how the privacy data can be handled in case of emergency.

| **Data security policy control by SMOOL SOFIA IoT platform** |
|---|
| **Related User Stories:**<br>• As a system operator, I want to apply at runtime security controls in the IoT system.<br>• As a system operator, I want to include to control data security policy transmitted between things (authentication, authorisation, confidentiality, integrity, non-repudiation) in the IoT system.<br>**Description:**<br>The enabler has insights of system communication *security metadata* and according to them it can perform automatic actions like filter data transmissions to and from devices not transmitting appropriate security metadata and sending alarms to the monitoring tool. |
| **Thing authorisation control by SMOOL SOFIA IoT platform** |
| **Related User Stories:**<br>• As a system operator, I want to apply at runtime security controls in the IoT system.<br>• As a system operator, I want to control when smart things are authorised to transmit data (specially in some critical situations, like in transmissions of actuation orders).<br>**Description:**<br>The enabler has insights of system communication *security metadata* and according to them it can decide to perform automatic actions like filter data transmissions to and from unauthorised devices and sending alarms to the monitoring tool. |

### 2.2.4. System Layer

#### *2.2.4.1. Test & Simulation*

| **Provide a production-like environment for developer to test their applications** |
|---|
| **Related User Stories:**<br>• As an IoT Application Developer, I want to simulate my SIS so that I can have a production-like environment to test my IoT application<br>**Description:**<br>Based on the system architecture and pre-collected behaviours logs, the Test & Simulation Enabler provides a simulation of the real SIS. The simulation system provides a production-like environment for developer to test their applications in some specific conditions without affecting the real SIS. |
| **Provide the ability to simulate the system in some specific situations** |
| **Related User Stories:**<br>• As an IoT Application Developer, I want to simulate my SIS in expected situation, so that I can test the reliability of my SIS.<br>**Description:**<br>The Test & Simulation Enabler supports simulating the system in some specific situations in which the developer has designed the test cases to verify the reliability of the system such as: functionality, usability, compatibility, operation, services. Every change in the system will be verified through all the test cases. The developer can quickly see the effects of the new features, new updates on the whole system. This is to ensure that the changes in the system will not affect the reliability of the system. |
| **Provide the ability to simulate some unexpected behaviours of a SIS** |
| **Related User Stories:**<br>• As an IoT Application Developer, I want to simulate some unexpected behaviours of my SIS, so that I can test the resilience of my SIS.<br>**Description:** |

The unexpected behaviours are not always happening in the real SIS. That why it is very important to simulate the unexpected situation so that the developer can make sure to handle all (most of) the corner-cases. The Test & Simulation Enabler can simulate some unexpected behaviours of the system such as: network issues, anomaly value, etc. And with the feedbacks from this simulation, the developer can improve the design of the SIS so that it can handle most of unexpected situations (resilience).

**Provide the ability to simulate some cyber-attacks on a simulated SIS**

**Related User Stories:**

- As an IoT Application Developer /Security Expert, I want to simulate some cyber-attacks so that I can test the security of my SIS

**Description:**
Due to the limit of resource, power and simple communication protocols, the IoT system has many vulnerabilities. The Test & Simulation Enabler provides the function to simulate some cyber-attacks such as: DoS, poison content, etc. By simulating the SIS in a cyber-attack, the developer can increase the security level of the SIS or prepare the solution to avoid such kind of attacks.

**Provide the ability to test the performance and the scalability of a SIS**

**Related User Stories:**

- As an IoT Application Developer, I want to simulate my SIS in a large scale, so that I can test the performance, the scalability of my SIS

**Description:**
Deploy a real SIS with a large number of devices and sensors, is not always easy because of the lack of equipment, the complexity of their installation, etc. The Test & Simulation Enabler supports to multiply the number of devices/sensors so that the performance, the scalability of the IoT system can be assessed easily and at low cost.

**Related User Stories:**

- As an IoT Application Developer, I want to trigger test automatically so that I can perform all the tests everytime there are something change in my SIS

**Description:**
Every change, every update can affect the trustworthiness of SIS. It can expose some vulnerabilities; it also can break some existing functionalities. To ensure the trustworthiness of the SIS, all the change and update should be tested. The Test and Simulation provides the ability to prepare the datasets, test cases, test campaigns and the trigger to tests automatically so that for every change and update, all the testing scenarios will be executed.

**Provide the ability to test the SIS with the recorded data from the real SIS**

**Related User Stories:**

- As an IoT Application Developer, I want to record the data from a real SIS, so that I can use the recorded data to test on my SIS

**Description:**
The generated datasets are not always accurate with the reality, and very often there are the case that the system does not behave as expected. So to investigate in such cases, the Test and Simulation provides the ability to record the data from the real SIS in some period of time, so the developer can use the recorded data to repeatedly testing the SIS to understand the behaviour of the real SIS.

### 2.2.5. Monitoring & Analytics Layer

#### *2.2.5.1. Security & Privacy Monitoring*

| **Near real-time monitoring of the overall SIS** |
|---|
| **Related User Stories:**<br>• As a system operator, I want to monitor the network traffic of the IoT system, so I can make sure the correct (secure) behaviour of the SIS is maintained, i.e. the availability, integrity and confidentiality of the data in transit are maintained.<br>• As a system operator, I want to monitor the multi-protocol network traffic of the IoT system (TCP/IP, Modbus TCP, MQTT, Wi-Fi, Z-wave, etc.) so I can apply security rules over the SIS communications to control the correct behaviour of the SIS.<br>• As a system operator, I want to monitor the log data generated by the devices' operating system managed by the IoT system, so I can apply security rules over the SIS level events to control the correct behaviour of the SIS.<br>• As a system operator, I want to get informed in real-time of network traffic events in the IoT system so as to have immediate and accurate information on the security-related events of the SIS.<br>**Description:**<br>The Security & Privacy Monitoring tool is responsible for collecting and storing a wide range of security-related data and enable the detection of multiple events: from general network events, to operating system process events in IoT devices, and (when SMOOL SOFIA IoT Platform is used in the SIS) application layer events.<br>All these events are displayed in HCIs to provide SIS operators with trends of data, warnings, alarms, or configuration changes. All of them displayed in user-friendly formats like sortable tables, time-series charts or geographical maps.<br>Still, the information and notification on detected events needs to be near real-time so as early reaction to detected incidents is allowed. |
| **Scalability for Security monitoring of SIS** |
| **Related User Stories:**<br>• As a system operator, I want my security monitoring deployed in the IoT system is scalable with the number of devices the IoT application may need to manage at a certain point of time.<br>• As a system operator, I want no latency or delays in received notifications or alarms when security incidents occur in my IoT system despite the number of IoT devices working in the system.<br>**Description:**<br>The Security & Privacy Monitoring tool is able to scale with large number of devices in the IoT system, so no monitoring performance reduction is suffered which may negatively impact the early feedback to Dev phase. |
| **Security Monitoring in SMOOL SOFIA IoT Platform** |
| **Related User Stories:**<br>• As a system operator, when I use SMOOL SOFIA IoT Platform, I want to ensure the security of the IoT application at runtime, so I can verify the correct (secure) behaviour of the SIS.<br>• As a system operator, I want to monitor the network traffic of the SMOOL SOFIA IoT Platform, so I can apply security rules over the application to control the correct behaviour of the SIS.<br>• As a system operator, I want to monitor the syslog data generated by the SMOOL SOFIA IoT Platform, so I can apply security rules over the application to control the correct behaviour of the SIS. |

- As a system operator, I want to have a set of default security incidents detection rules to be selected and applied over the SMOOL SOFIA IoT Platform, so I can apply well-known incidents detection rules over the SIS.
- As a system operator, I want to check whether the communication is encrypted, so that I can know if the application layer communication is secure.
- As a system operator, I want to check whether the application user ID is transmitted in clear, so that I can know if the application layer communication is secure.

**Description:**
The Security & Privacy Monitoring tool needs to be integrated with the Security features offered by the enhanced IoT Platform SMOOL, in order to be able to monitor availability, confidentiality, integrity and access control of data in transit between the IoT devices connected to the IoT Platform. The Security & Privacy Monitoring tool is responsible for collecting and monitoring security-related data of application layer when SMOOL SOFIA IoT Platform is used, so as notifications of events related to these data are generated.

**Advanced anomaly detection of SIS based on AI**

**Related User Stories:**

- As a system operator, I want to ensure the security of the IoT system at runtime, so I can verify the correct (secure) behaviour of the SIS.
- As a system operator, I want to ensure that any incident that does not classify in the normal behavioural pattern of the SIS is notified.

**Description:**
The Security & Privacy Monitoring tool is able to detect anomalies on the SIS behaviour which may indicate symptoms of security incidents or external attacks, or at least are events beyond the standard behavioural patterns of the SIS. The anomaly detection capability can detect three main types of groups of anomalies: (i) asset discovery related anomaly events, (ii) IDS data related anomaly events and (iii) protocol specific related anomaly events.

**Situational awareness about SIS security**

**Related User Stories:**

- As a system operator, I want to ensure the security of the IoT system at runtime, so I can verify the correct (secure) behaviour of the SIS.
- As a system operator, I want to visualize in a well-structured, friendly and easy to understand way the security related information of the SIS, including all real time events and alarms, so I can understand the security situation of the SIS and verify the correct status.

**Description:**
The Security & Privacy Monitoring tool is able to provide holistic situational awareness about the SIS security correlating information from multiple sources in the SIS.

#### *2.2.5.2. Root Cause Analysis*

**Determine the root cause of operational issues**

**Related User Stories:**

- As an IoT DevOps engineer, I see that the monitoring tools of the platform are reporting failures, and I would like to know what the root cause of the reported issues.
- As an IoT DevOps engineer, I witness an incident and want to save my experience in dealing with it so that when it is repeated, I can react more quickly and more accurately.
- As an IoT DevOps engineer, I know some potential vulnerabilities in my system and want to react as early as possible if someone exploits these vulnerabilities.

**Description:**
The Root Cause Analysis engine implements algorithms to correlate data from multiple points and determine the most-likely root cause of a reported issue. Ranging from purely operational to security and privacy issues, the final root case of detected misbehaviour can be hidden in complex relationships between dates spread across multiple components. The Root Case Analysis engine provides a core that performs such correlation of data and gives a list of the possible origin if the observed misbehaviour. Instead of relying on human experts to exhaust all the causal connections between incidents and symptoms, the RCA tool builds this knowledge itself, by recording the typical incidents and their symptoms. During runtime, it compares the similarity between the observed symptoms with the recorded ones in the library to identify the possible incidents.

**Assess the risk analysis and impact of the detected operational issues**

**Related User Stories:**

- As an IoT DevOps engineer, knowing the root cause of an ongoing issue helps me to correctly assess the countermeasures (even if triggered automatically) and the impact level of the detected malfunctions of the system.

**Description:**
Once a problem has been detected, the Root Cause analysis engine will compute the most-probable source of the observed issue. This is not an isolated output of the component, but a crucial data that is being used by the DevOps engineer to correctly assess the countermeasures and the potential impact (both operational and economical) of the detected issue.

**Enhance the prediction capabilities of the system using previous data about failures.**

**Related User Stories:**

- As an IoT DevOps engineer, I would like the system to continuously learn about the previously detected failures by using the root cause analysis, test and simulation and online learning engines.
- As a system operator, I would like to see the whole evolution of the system from a normal state to a misbehaviour so that I can predict the failures before there is eventually significant damage.

**Description:**
Root Cause Analysis tool is also used as the input of other automated components of the ENACT platform. Due to the nature of the process performed by this component, its output enriches the reasoning performed by other components of the platform, such as the test and emulation, and the online learning engines. In this sense, the results generated by the root cause analysis engine help the DevOps engineer to accelerate the development process by predicting already-known failures and their respective root cause.

#### *2.2.5.3. Behavioural Drift Analysis*

**Behavioural drift model specification**

**Related User Stories:**

- As an IoT DevOps engineer, I need to specify models of what SIS behaviours I expect to observe in the physical environment.
- As an IoT DevOps engineer, I want to use a graphical designer to input a probabilistic observation model into the observation engine, so that it is possible to configure how the application is being monitored.
- As an IoT DevOps engineer, I want to design the model to represent the system states and characteristics.[7]
- As an IoT DevOps engineer, I want to set the values on each transition of the previously designed model, so that I can parametrize the model.

[7]The original user story was: As an IoT DevOps engineer, I want to design the HMM (Hidden Markov Model) state graph for that model to represent the system states, so that I can define the model used to compute behavioural drift.

- As an IoT DevOps engineer, I want to set the values on each state of the previously designed, so that I can parametrize the model.
- As an IoT DevOps engineer, I want a way to export the designer's output into the observation engine, so that my model can be used by the observation engine to compute behavioural drift in the application.

**Description:**
Behavioural Drift Analysis is to detect whenever the application derives from its expected behaviour and monitor such drifts. The developer specifies the model of the expected behaviour in the physical environment. He describes a probabilistic behavioural model based on a deterministic Finite State Machine (FSM) as one expected observed behaviour and adds uncertainties models on inputs/outputs occurrences.

**Probes and context monitoring**

**Related User Stories:**

- As a system operator, I want to be able to monitor the behavioural drift of my applications once deployed, so that I can ensure that they function appropriately.
- As a system operator, I want to install an observation engine to measure the behavioural drift of one application, so that I allow system operators to monitor it specifically
- As a system operator, I want to have access to the drift measurements for all my applications, so that I am able to monitor a whole installation comprised of multiple applications.[8]
- As an IoT DevOps engineer, I need help to collect and synchronise contextual data to observe the running SIS.

**Description:**
Enabler algorithms collect contextual data from sensors and probes in the system and produce a continuous stream of synchronized contextual information for monitoring the behaviour of SIS. Finally, we do not provide the system operator with a predefined dashboard, but rather provide drift measurements through an MQTT broker, so that he can specify the dashboard adapted to his needs.

**Behavioural drift measure**

**Related User Stories:**

- As an IoT DevOps engineer, I see that the observed behaviour of the running system is drifting from the expected one.

**Description:**
Behavioural drifts analyser is based on a set of observers that compute one measure between 0 and 1 to evaluate the difference between the expected behaviours and the observed ones (1 is a perfect conformance between them). For the designer to easily interact with the behavioural drift measurement tool, the description of the probabilistic behavioural model is based on a deterministic Finite State Machine (FSM) describing the ideal expected behaviour and added uncertainties models on inputs/outputs occurrences.

**Behavioural drift analysis: description of the observed behaviour**

**Related User Stories:**

- As a system operator, I want to be able to get a vision of the most likely model for the real-world application derived from the stochastic model of the observed application, so that I am able to detect anomalies.
- As an IoT DevOps engineer, I see that the observed behaviour of the system deviates from the expected behaviour, and I would like to interpret the symptoms of the observed drift.

**Description:**

[8] The origical user story was: As a system operator, I want to have access to a central dashboard regrouping the drift measurements for all my applications, so that I am able to monitor a whole installation comprised of multiple applications.

Behavioural Drift Analysis is to detect whenever the application derives from its expected behaviour. A deterministic Finite State Machine (FSM) is describing the ideal expected behaviour, supposed to be known a priori. Developers are more familiar in working with FSM rather than complex stochastic models that include uncertainties modelling through complex distributions. The behavioural drift analysis presents to the developer a more comprehensive model (as a difference from the specified model) to understand the symptoms of the observed and measured drift.

Within the next section we showcase how ENACT enablers can be integrated with, by listing the available API endpoints and their responsibility. Many of these endpoints are already used by cross enabler integration or / and integration with 3rd party tools listed in the Section 5.

## 2.3. API's description for integration

This section builds upon the API's integrations listed in the deliverable 5.2 with extension of the final set of API calls and descriptions available at the time of the final release of the project results.

### 2.3.1. Risk Management

In the following we provide details about the API exposed by Risk Management that enable the integration of enabler with other tools. A set of high-level commands are exposed by Risk Management in the form of a REST API. This includes commands to retrieve and provide a architectural and dataflow model, vulnerabilities, risks and mitigation actions.

| Method | Resource | Content-type | Description |
|---|---|---|---|
| **GET** | /graphql=query | JSON | GraphQL to fetch the project assets, vulnerabilities, treatments and risks |
| **POST** | /api/risk | JSON | Add Risk |
| **POST** | /api/vulnerability | JSON | Add Vulnerability |

Example of a GraphQL query to list all the objects:

```
{
 containers(where: { path: "RootId", comparison: "equal", value: "$id" }) {
  name
  bpmn
  payload
  riskStatus {
   icon
   name
   color
   statusId
  }
  assets {
   id
   name
   payload
   group
   evidences {
    id
    name
   }
   vulnerabilities {
    id
    name
   }
   risks {
    id
    name
    payload {
     stride
     lindun
     impact
     likelihood
    }
    vulnerabilities {
     rootId
```

### 2.3.2. Online Learning

Two interfaces can be used to integrate the Online Learning tool with other Enablers/tools: a Web API with corresponding commands and communication via MQTT.

| Method | Resource | Content-type | Description |
|---|---|---|---|
| **POST** | /api/submit_configuration | JSON | This endpoint is used to modify the configuration of the algorithm. This includes (1) the type of the algorithm, (2) observation-space and action-space, and (3) |

| | | | |
|---|---|---|---|
| | | | the hyperparameter of the chosen algorithm |
| **POST** | /api/get_action | JSON | This endpoint is used to transmit the current state variables and the last reward to the algorithm. The response of the request contains the action to be executed. |
| **GET** | /api/get_environmental_log_data | text | This endpoint is used to obtain logs of the learning process. The query string can contain the parameter first_timestep, which causes only logs after this timestep to be returned. |
| **GET** | /api/external_data/ | text | External data, for example from a connected tool, can be obtained by a GET request. For this purpose, as described above, the query string can contain the parameter first_timestep. |
| **POST** | /api/external_data/ | JSON | This endpoint can be used to transmit data of external sensors. |
| **GET** | /api/annotations | JSON | This endpoint can be used in two ways. Firstly, all diagram annotations can be obtained. |
| **POST** | /api/annotations | JSON | New annotations can be transmitted. |
| **POST** | /api/upload_model | JSON with Base64 string | This endpoint is used to load an existing model and its configuration into the tool. To do so the ZIP file generated by the save() method of Stable-Baselines 3 must be POSTed as a Base64 encoded string inside of a JSON. |

*Table 1. High-level commands of OLE Web API.*

At each time step the observations are published in the MQTT-Broker. These can be found under topic: <use-case name>/observations/<observation variable>. Data published from the outside can be displayed in the GUI.

In the following we details the interaction between OLE and other ENACT tools.

#### 2.3.2.1. *Online Learning as input provider for Actuation Conflict Management*

The Actuation Conflict Management can be used to deploy conflict management nodes based on several main operating principles, like ECA rules or Boolean logic. A third possible principle is to solve the conflict generated from the Online Learning enabler.

| | |
|---|---|
| Data to be exchanged: | Off-the-shelf actuation conflict manager strategy: a description of what the strategy does along with implementation and/or configuration details required to instantiate the actuation conflict management component. |
| Format: | Off-the-shelf actuation conflict manager strategy: JSON |
| Communication protocol: | REST over HTTP |
| Rationale | Results from Online Learning are then fed back into the actuation conflict manager off-the-shelf strategy database to be later deployed in the application. |

### 2.3.3. Things behaviour modelling framework

ThingML is an open source IoT framework that includes a language and a set of generators to support the modelling of system behaviours and their automatic derivation across heterogeneous and distributed devices on the IoT and edge spaces. The ThingML code generation framework has been used to generate code in different languages, targeting around 10 different target platforms (ranging from tiny 8 bit microcontrollers to servers) and 10 different communication protocols. ThingML models can be platform specific, meaning that they can only be used to generate code for a specific platform (for instance to exploit some specificities of the platform); or they can be platform independent, meaning that they can be used to generate code in different languages. In the context of ENACT, ThingML has been extended with a logging mechanism, which provide a means to remotely monitor the execution flow of a ThingML program. This is particularly relevant for debugging resource constrained devices such as Arduinos. Following the ThingML philosophy, the logging mechanism is platform independent, meaning the logging of several software implemented using different languages deployed on different target is reified into a single abstraction level as ThingML concepts.

There are two main approaches for integrating ThingML with development, building and deployment solutions:

1. Using the ThingML CLI as detailed below:

```
Typical usages:
    java -jar your-jar.jar -t <tool> -s <source> [-o <output-dir>] [--
options <option>][-d]
Usage: <main class> [options]
  Options:
    --compiler, -c
      Compiler ID (Mandatory unless --tool (-t) is used)
    --create-dir, -d
      Create a new directory named after the configuration for the output
      Default: false
    --help, -h
      Display this message.
      Default: false
    --list-plugins
      Display the list of available plugins
      Default: false
    --options
      additional options for ThingML tools.
    --output, -o
      Optional output directory - by default current directory is used
    --source, -s
```

```
    A thingml file to compile (should include at least one configuration)
  --tool, -t
    Tool ID (Mandatory unless --compiler (-c) is used)
```

2. **Using ThingML as a library into a project source code.** The listing below describes how to programmatically process ThingML models.

```
File myFile = new File("source.thingml");
ThingMLModel myModel = ThingMLCompiler.loadModel(myFile);
//Do something
ThingMLCompiler.saveAsThingML(myModel, "target.thingml");
//or
ThingMLCompiler.saveAsXMI(myModel, "target.xmi");
```

The ThingML library can be included into a project as a Maven dependency as described below.

```
<dependency>
    <groupId>org.thingml</groupId>
    <artifactId>compilers.registry</artifactId>
    <version>2.0.0-SNAPSHOT</version>
</dependency>

...

<repository>
   <id>thingml-snapshot</id>
   <name>thingml-snapshot</name>
   <url>http://maven.thingml.org/thingml-snapshot/</url>
</repository>

<repository>
   <id>thingml-release</id>
   <name>thingml-release</name>
   <url>http://maven.thingml.org/thingml-release/</url>
</repository>
```

### 2.3.4. GeneSIS

In the following we provide details about the API exposed by GeneSIS that enable the integration of GeneSIS with other tools (from ENACT or not). A set of high-level commands are exposed by GeneSIS in the form of a REST API. This includes commands to retrieve and provide a GeneSIS model.

| Method | Resource | Content-type | Description |
|---|---|---|---|
| **GET** | /genesis/types | Response: application/json | To retrieve all the component types registered in the execution engine |
| **POST** | /genesis/deploy | Response: application/json<br>Parameter: application/json | To provide the engine with a new deployment model and trigger a re-deployment |

| | | | |
|---|---|---|---|
| **GET** | /genesis/logs | Response: text/plain | To retrieve all the logs from the execution engine |
| **GET** | /genesis/model | Response: application/json | To retrieve the current deployment model |
| **GET** | /genesis/model_ui | Response: application/json | To retrieve the current deployment model including its graphical representation |
| **POST** | /genesis/push_model | Response: application/json | To push a new target deployment model into the GeneSIS deployment engine without triggering the deployment |
| **POST** | /genesis/deploy_model | Response: application/json Parameter: application/json | If a target deployment model is loaded in the execution engine and has not been deployed yet, deploys it |
| **GET** | /genesis/get_target_model | Response: text/plain | To retrieve the target deployment model |
| **POST** | /api-docs | Response: Application/xml | Generates and delivers the API documentation |

*Table 2. High-level commands.*

One of the objectives for the second period was to extend the GeneSIS execution engine façade with an API enabling the MOF (Meta Object Facility) reflection atomic modifications of a deployment model. This includes introspecting and modifying components and links as well as adding/removing components and links into a deployment model. This interface is typically used to update a deployment model, e.g. to update the link to the software to be deployed or a version number.

| Method | Resource | Content-type | Description |
|---|---|---|---|
| **PUT** | /genesis/component | Response: application/json Parameter: application/json | To update an attribute of a component or add a component. Consumes as parameter, a JSON including, the name of the component, the name of the attribute and the new value or the JSON serialization of a whole component in order to add it to the deployment model |

| | | | |
|---|---|---|---|
| **PUT** | /genesis/link | Response: application/json<br>Parameter: application/json | To update an attribute of a link or add a link. Consumes as parameter, a JSON including, the name of the link, the name of the attribute and the new value or the JSON serialization of a whole link in order to add it to the deployment model |
| **GET** | /genesis/component | Response: application/json | To retrieve information about a specific component. Return a JSON description of the component, including runtime information (i.e., status) |
| **GET** | /genesis/link | Response: application/json | To retrieve information about a specific link. Returns a JSON description of the link |

*Table 3. GeneSIS Model manipulation API*

In the following we details the interaction between GeneSIS and other ENACT tools.

#### 2.3.4.1. GeneSIS as input provider for Risk Management

To understand the implications and assess the risk of the current IoT architecture, risk analysis process needs to take place. Usually, the risk management process is performed against singular component of the application. Within ENACT risk management process, we can graphically represent the IoT architecture by reading the GeneSIS model which contains the architectural description, additional technical attributes and even the dependencies between the components of the architecture. This allows the risk management tool to perform the risk analysis process against single component, multiple connections which together pose a risky situation or even a risky components connection.

| | |
|---|---|
| Data to be exchanged: | Deployment model: a deployment model containing the SIS. |
| Format: | JSON |
| Communication protocol: | File based GeneSIS export or REST over HTTP |
| Rationale: | The Risk Management tool takes into account the architectural description in order to provide the basic means required for architectural based risk assessment process. By knowing the architectural model before the risk analysis process is started, additional risky situations can be detected / analysed and mitigated. |

#### 2.3.4.2. GeneSIS as input provider for Actuation Conflict Management

The Actuation Conflict Management (ACM) analyses a GeneSIS deployment model to identify conflicts and helps DevOps engineers solving these conflicts. The GeneSIS model contains references to applications to deploy and the ACM uses the deployment model to address all the deployed applications at one time.

| | |
|---|---|
| Data to be exchanged: | Deployment model: a deployment model containing the application(s). |

| | |
|---|---|
| Format: | Deployment model: JSON |
| Communication protocol: | Manual and/or REST over HTTP |
| Rationale | The ACM consumes a GeneSIS deployment model to identify conflicts. Since GeneSIS is the tool used to deploy applications, and that for the ACM to efficiently detect conflicts it needs to reason on the entire application, it needs a full model describing links between all things as input. |

#### 2.3.4.3. GeneSIS as input provider for DivEnact

| | |
|---|---|
| Data to be exchanged: | Deployment model: a deployment model containing the SIS. |
| Format: | Deployment model: JSON |
| Communication protocol: | Manual and/or REST over HTTP |
| Rationale | The diversifier at design time takes as input a GeneSIS model that specifies the architecture (topology) of an IoT system, and generate multiple GeneSIS models specifying diverse architecture<br>DivEnact takes as input a GeneSIS model that describes the deployment plan for a local system, and deploy it into multiple local systems in a fleet by invoking the API of the remote GeneSIS engine on the main edge device of each local system. |

#### 2.3.4.4. GeneSIS as input provider for test and simulation environment

Test and Simulation tool requires the architectural components description in order to understand what components of the architecture can be simulated / recorded and then played back. GeneSIS carries majority of the definition required that are a baseline for Test and Simulation tool in order to record the effective traffic, model it and possibly replay the test scenarios.

| | |
|---|---|
| Data to be exchanged: | Deployment model: a deployment model containing the SIS. |
| Format: | JSON |
| Communication protocol: | File based GeneSIS export |
| Rationale: | The Test and Simulation tool can run predefined test scenarios against the application deployed on a test environment where the IoT devices, their inputs and outputs are simulated. Knowing which components can be simulated and where the data is sent / received is crucial for the simulation and it in majority provided by the GeneSIS model. |

In addition, GeneSIS can be used trigger automatically a test campaign via the test and simulation enabler. To do so, GeneSIS simply trigger a HTTP GET request to the test and simulation enabler once a deployment is completed.

| | |
|---|---|
| Data to be exchanged: | None |
| Format: | N/A |
| Communication protocol: | HTTP GET request |
| Rationale: | GeneSIS simply produce a HTTP GET request to the test and simulation enabler once a deployment is completed thus triggering a test campaign. This mechanism enable the full integration of GeneSIS and the test and simulation enabler within a proper DevOps pipeline. |

#### 2.3.4.5. GeneSIS as input provider for security & privacy monitoring

The Security & Privacy monitoring tool requires information about the security and privacy controls that have been actually deployed in the SIS in order to monitor the performance of the mentioned controls together with all the deployed components of the SIS.

| | |
|---|---|
| Data to be exchanged: | Deployment Model: information related with security and privacy controls deployed on the SIS. |
| Format: | Deployment model: JSON |
| Communication protocol: | REST over HTTP |
| Rationale | The GeneSIS module will be responsible of deploying different components of the SIS and will also deploy security and privacy controls required to protect the SIS. This information will be used by the Security & Privacy Monitoring tool to know exactly which security and privacy controls are deployed and some of the detected events will be potentially linked to misbehaviour or bad performance of the deployed controls |

#### 2.3.4.6. GeneSIS as input provider for Root cause analysis

The Root Cause Analysis (RCA) engine requires crucial information from the SIS component in order to detect potential misbehaviour and determine its root cause. In particular, the RCA will require to access the list of deployed devices, in order to know what the topology of the network is, and access to the monitoring probes that will be used to compute the actual state of the network. Both inputs will be used to construct the system's graph and detect possible misbehaviours on the monitored devices.

| | |
|---|---|
| Data to be exchanged: | • Deployment Model: information related with devices deployed on the network.<br>• RCA Monitoring probes: Information about all the deployed nodes with the instrumentation capabilities, in order to extract suitable information for the RCA. |
| Format: | • Deployment model: JSON<br>• Monitoring Probes: log files, monitoring devices outputs, etc. |
| Communication protocol: | REST over HTTP |
| Rationale | The GeneSIS module will be one of the main inputs of the RCA component, since the latter needs to build the actual system graph. Using information about the topology of the network the RCA module will be aware of the devices that are being used. Moreover, the RCA will also receive information about the instrumented devices on which the RCA can rely to extract information. Both types of information will be used in the RCA to construct the actual system graph, detect potential malfunctions and compare these scenarios with the pre-defined database. |

### 2.3.5. Actuation Conflict Management

#### 2.3.5.1. The high-level commands API

A set of high-level commands are exposed by the Actuation Conflict Management enabler in the form of a REST API. This includes commands to retrieve and provide a WIMAC (Workflow and Interaction Model for Actuation Conflict management) model.

| Method | Resource | Content-type | Description |
|---|---|---|---|
| POST | /acm-renderer/model | Parameter: application/json<br>Response: application/json | Generates a WIMAC from a service address, a service type and an environment model, along with its rendering for the view |
| POST | /acm-model-editor/loadFile | Parameter: application/json<br>Response: application/json | Generates a WIMAC from a raw model and service type then renders it |
| POST | /acm-renderer/updateEnvModel | Parameter: application/json<br>Response: application/json | Updates the environment model in the given WIMAC, and sends out a rendered model of it for the user interface |
| POST | /acm-renderer/deployOnline | Parameter: application/json<br>Response: application/json | Deploys a WIMAC to a target, firstly generating the deployment model then sending it to the appropriate application for immediate use |
| POST | /acm-renderer/deployDownload | Parameter: application/json<br>Response: application/json | Downloads the deployment model that would have been deployed to the application |

These commands allow usage of the Actuation Conflict Management as a gateway to and from WIMAC (see D2.3 for more details), letting other software leverage the abstractions provided by WIMAC including the environment model. Further information including data types can be found in the documentation available at /api-docs/.

#### 2.3.5.2. The model manipulation API

This set of commands are related to manipulating WIMACs to identify and solve conflicts. Note that findConflicts is a pre-requisite before calling any of the following commands including solving all conflicts with default components. Further information including data types can be found in the documentation available at /api-docs/.

| Method | Resource | Content-type | Description |
|---|---|---|---|
| POST | /acm-renderer/findConflicts | Parameter: application/json Response: application/json | Finds conflicts in a WIMAC model |
| POST | /acm-renderer/solveConflictsDefault | Parameter: application/json Response: application/json | Solves all conflicts detected in the WIMAC using the default strategy |
| POST | /acm-renderer/select | Parameter: application/json Response: application/json | Returns the off the shelf ACM strategies pertaining to a conflict |
| POST | /acm-renderer/instantiateMonitor | Parameter: application/json Response: application/json | Takes a conflict ID and a strategy, instantiates the Monitor node in the WIMAC |
| POST | /acm-renderer/instantiateACM | Parameter: application/json Response: application/json | Takes a conflict ID and a strategy, instantiates the ACM node in the WIMAC |

#### 2.3.5.3. The actuation conflict manager API

These commands enable the consultation and enrichment of the strategy database, as well as validation for customized ECA+ behaviours. Further information including data types can be found in the documentation available at /api-docs/.

| Method | Resource | Content-type | Description |
|---|---|---|---|
| POST | /acm-database/addStrategy | Parameter: application/json Response: application/json | Adds a new off the shelf ACM strategy to the database |
| GET | /acm-database/strategyDatabaseListing | Response: application/json | Lists all strategies and their information available in the database |
| POST | /acm-database /queryStrategyDatabase | Parameter: application/json Response: application/json | Returns a filtered list of strategies based on a query, which can be SPARQL (for Jena) or metadata (for both Jena and Rethinkdb) |
| GET | /acm-database/queryDatabaseStatus | Response: application/json | Check whether or not the database is initialized properly |

| | | | |
|---|---|---|---|
| POST | /acm-model-editor/verifyECA | Parameter: application/json<br>Response: application/json | Validates an ECA model, putting it through all the model checking steps |

#### 2.3.5.4. The actuation conflict detection tuning API

This command can be used to tune the rules used by the AGG (Attributed Graph Grammar) engine to transform the WIMAC model.

| Method | Resource | Content-type | Description |
|---|---|---|---|
| POST | /acm-model-editor/editAGGRules | Response: application/json | Launches AGG-Editor for ACM conflict finding rule edition |

In the following we details the interaction between ACM and other ENACT tools.

#### 2.3.5.5. Actuation Conflict Management as input provider for Online Learning

The Online Learning tool needs information about the action space beforehand. Based on the size of the action space the actions coming from the algorithm needs to be clipped to fit for the current system to adapt. These information needs to be provided either by GeneSIS or ACM, depending on which tool is responsible for the actual execution of an action.

| | |
|---|---|
| Data to be exchanged: | Action space (e.g., number of parameters that should be learned (currently only one parameter per instance of the online-learning tool supported) & dimensions of the parameter (e.g., 0-1)) |
| Format: | JSON |
| Communication protocol: | Manual and/or REST over HTTP |
| Rationale | Online Learning proposes new parameter values for the software systems whose adaptation logic should be optimized. As the parameter is interpreted as the action variable, OL needs to know the boundaries of the parameter beforehand. |

#### 2.3.5.6. Actuation Conflict Management as input for GeneSIS

Once actuation conflicts are identified, the actuation conflict Management support DevOps engineers in either selecting off-the-shelf actuation conflict manager or in designing new ones to be deployed as part of the SIS. Once the developer has ended his job, a GeneSIS model is generated to be deployed to replace the existing applications, with solved conflicts.

| | |
|---|---|
| Data to be exchanged: | • Deployment model: a deployment model including the actuation conflict managers and how to deploy them, their relationships with the other components in the model, the constraints on these relationships (i.e., all the communication to an actuator should transit via the actuation conflict management).<br>• Actuation conflict management: the implementation of the actuation conflict manager needs to be generated and provided to GeneSIS. |
| Format: | • Deployment model: JSON<br>• Actuation conflict management: binary, ThingML code, etc. |
| Communication protocol: | Manual and/or REST over HTTP |
| Rationale | For GeneSIS to properly deploy a new version of the SIS including the actuation conflict management, it needs information about (i) where the actuation conflict management should be located in the overall architecture of the system (i.e., where it should be deployed and its interactions with the rest of the system) and (ii) the actual implementation of the actuation conflict manager to be deployed. GeneSIS already embed the necessary concepts to deploy actuation conflict managers (in particular with the, control relationship between components see D2.2). It is worth |

| | |
|---|---|
| | noting that GeneSIS is not bound to actuation conflict managers generated from the Actuation Conflict Management. |

### 2.3.6. DivEnact

#### *2.3.6.1. DivEnact as input for GeneSIS*

GeneSIS manages the deployment of a local SIS, typically used by a particular customer. The ENACT diversity controller at run-time is in charge of managing the fleet of a large number of such local sub systems and control the emerging or engineered diversity among these systems.

| | |
|---|---|
| Data to be exchanged: | • GeneSIS deployment model.<br>• GeneSIS component update and implementation. |
| Format: | JSON |
| Communication protocol: | REST over HTTP |
| Rationale | DivEnact engine deploys a GeneSIS engine on the central edge device of each local system, and after that, during the whole lifecycle of the entire fleet, manages the local deployment of these sub systems remotely and automatically. In particular, when updates or patching is required, it selects the relevant edge devices, identifies the particular and customised deployment model (GeneSIS model) for each sub system, and sends these models to the GeneSIS engines by invoking its REST API. The latter will enact the deployment on the local systems. In the other direction, developers who use GeneSIS to define the deployment model for one sub system can call DivEnact to deploy the current deployment into the fleet of sub systems, by invoking DivEnact's REST APIs as defined below. |

DivEnact provides a public REST API, comprising the following methods. DivEnact's default web GUI uses this APIs to convert the GUI operations into deployment actions in the backend. In addition, the API provides external tools a programmable way to interact with the DivEnact engine. Within the project, GeneSIS uses this API to release (or preview) local deployment model to a fleet. In addition, the public API allows DevOps teams to integrate DivEnact into their continuous delivery pipelines.

The main part of the API is the group of three "global" methods, which supports the release, preview and diversification operations in the fleet level. The "deployment" and "device" methods are used to directly manipulate the deployment models and devices in the fleet. The "template" and "variant" methods are used to handle the DivEnact concepts to generate multiple deployment models from standard templates.

| Method | Resource | Content-type | Description |
|---|---|---|---|
| PUT | /global/production/:variant | Response: application/json | Deploy one variant as the production version to all relevant devices |
| PUT | /global/preview/:variant | Parameter: n: number<br>Response: application/json | Deploy the variant into n devices as a preview |
| PUT | /global/shuffle | Parameter: application/json | Deploy multiple variants in the parameter to all the devices, and keep the software diversity among the devices |
| GET | /deployment/get | Response: application/json | List all the deployments in the fleet |
| GET | /deployment/:deployment /applied | Response: application/json | Get a list of devices where the specified :deployment is applied. |

| | | | |
|---|---|---|---|
| PUT | /device/:device | Parameter: application/json | Attached the specified tags (as in the parameter) to the :device |
| GET | /variant/get | Response: application/json | List all variants |
| PUT | /variant/:variant | Parameter: :template, application/json | Create a new variant by resolving the variables with provided values (all in the json body) in the :template |
| DELETE | /variant/:variant | Response: application/json | Remove the :variant |
| GET | /template/get | Response: application/json | List all templates |
| PUT | /template/:template | Parameter: application/json | Create the :template using content in the json body |
| DELETE | /template/:template | Response: application/json | Remove the :template |

### 2.3.7. Test & Simulation

#### *2.3.7.1. Test & Simulation APIs for integration*

The Test and Simulation provides some APIs to integrate into any DevOps cycle.

| Method | Resource | Content-type | Description |
|---|---|---|---|
| GET | /devops | Response: application/json | Get automation testing configuration |
| POST | /devops | Parameter: application/json Response: application/json | Update the campaign and (or) webhookURL |
| GET | /devops/start | | Trigger the simulation and test process |
| GET | /devops/stop | | Stop the simulation and test process |
| GET | /devops/status | Response: application/json | Get the status of current execution |

In the following we details the interaction between Test and Simulation Enabler and other ENACT tools.

#### *2.3.7.2. Test & Simulation as input provider for Risk Management*

Test and Simulation Enabler can be used in context of reasoning on the likelihood and impact on the environment of the specific risk. The Risk Management Enabler is ought to send the request of running a specific tangible risk scenario against the simulated IoT environment in order to understand the risk implications. This in effect reassures the Risk Management actor about the correctness of the level of risk assessment.

| | |
|---|---|
| Data to be exchanged: | Risk to be analysed on the test scenarios |
| Format: | JSON |
| Communication protocol: | REST over HTTP |
| Rationale: | The Risk Management tool should provide means to validate the levels of likelihood and consequence values set by the user. By allowing the risk to be tested against the simulation it adds another level of validation to the analysis. |

#### 2.3.7.3. Test & Simulation and GeneSIS

Test and Simulation can be integrated with GeneSIS to perform the automation testing for every deployment.

| | |
|---|---|
| Data to be exchanged: | None |
| Format: | None |
| Communication protocol: | REST over HTTP |
| Rationale: | The GeneSIS send a simple GET request to the endpoint to trigger the Test and Simulation process automatically. |

When the Test and Simulation process is finished, the evaluation results are sent back to GeneSIS

| | |
|---|---|
| Data to be exchanged: | The evaluation results of all simulations |
| Format: | JSON |
| Communication protocol: | REST over HTTP |
| Rationale: | The Test and Simulation send back to GeneSIS the evaluation results of all the simulations, the GeneSIS can have different reactions based on the evaluation results. |

### 2.3.8. Context Monitoring and Behavioural Drift

A set of high-level commands are exposed by Context Monitoring and Behavioural Drift in the form of a REST API. This includes commands to retrieve and provide a model.

#### 2.3.8.1. The high-level commands API

| Method | Resource | Content-type | Description |
|---|---|---|---|
| POST | /bda-server/scxml/import | Parameter: application/json Response: application/json | Loads the SCXML file and converts it to a format suitable for manipulation by the Configuration tool |
| POST | /bda-server/scxml/export | Parameter: application/json Response: application/json | Exports the FSM being edited as SCXML |
| POST | /bda-server/genesis/export | Parameter: application/json Response: application/json | Generates a GeneSIS deployment model containing the required components to run Behavioural Drift Computation on the application |
| POST | /bda-server/genesis/deploy | Parameter: application/json Response: application/json | Generates a Node-RED deployment model containing the required components to run Behavioural Drift Computation on the application |

The scxml commands allow loading and saving application models. The genesis commands perform generation of deployment models containing the necessary components to deploy a configured behavioural drift computation application.

#### 2.3.8.2. The analysis API

| Method | Resource | Content-type | Description |
|---|---|---|---|
| POST | /bda-server/analysis/dot | Parameter: text/plain Response: application/json | Loads the dot file that will be used by the analysis tool as base model to work on |
| POST | /bda-server/analysis/observations | Parameter: text/plain Response: application/json | Loads the all-in-one CSV file to be used by the analysis tool in conjunction with the dot file |

| | | | |
|---|---|---|---|
| POST | /bda-server/analysis/run | Parameter: application/json<br>Response: application/json | Runs the analysis tool on the files provided by the previous analysis methods. This takes time. |

The dot and observations commands load data for the analysis. Note that the analysis takes a long time to complete.

In the following we details the interaction between B DA and other ENACT tools.

#### 2.3.8.3. Context Monitoring and Behavioural Drift Analysis as input provider for GeneSIS

In order to identify and analyse a drift in the behaviour of a SIS, the Behavioural drift analysis enablers needs to gather data from the SIS itself and from its environment. Monitoring probes are thus required to be deployed together with the SIS. Whilst they are defined and provided by the Behavioural Drift Analysis enabler, they are deployed by GeneSIS.

| | |
|---|---|
| Data to be exchanged: | • Deployment model: a deployment model including the necessary probes and analyser to monitor the SIS's context and behaviour.<br>• Monitoring Probes and Analyser: the implementation of the monitoring probes and analyser needs to be generated and provided to GeneSIS. |
| Format: | • Deployment model: JSON<br>• Monitoring Probes and Analyser: binary, ThingML code, etc. |
| Communication protocol: | Manual and/or REST over HTTP |
| Rationale | For GeneSIS to properly deploy a new version of the SIS including the monitoring probes and analyser, it needs information about (i) where they should be located in the overall architecture of the system (i.e., where it should be deployed and its interactions with the rest of the system) and (ii) their actual implementation. |

#### 2.3.8.4. Context Monitoring and Behavioural Drift Analysis as input provider for Online Learning

The Online Learning tool exposes an API to display external variables in the monitoring pane. As the behavioural drift can be computed by the BDA tool with the same information used by the Online Learning tool, it can be used to support the guidance of the learning process. To do so, the BDA tool can access the information provided by the OLE via MQTT and compute the behavioural drift value. The computed value is then provided via MQTT to Online Learning.

| | |
|---|---|
| Data to be exchanged: | • State & action variables (e.g., User requests of a web server, CPU-load, etc.)<br>• Metrics (e.g., average latency, max. latency of user requests)<br>• Behavioural drift value |
| Format: | JSON |
| Communication protocol: | MQTT |
| Rationale | As Online Learning should propose a parameter setting for a system, based on the current environment state, this state needs to be continuously monitored and forwarded to the OL-tool. Behavioural Drift value can be used as an additional metric to guide the learning process of the Online Learning Enabler. |

### 2.3.9. Security & Privacy Monitoring and Control

In the following we provide details about the interface exposed by the Security and Privacy Monitoring that enables the integration of the enabler with other tools. The Security and Privacy Monitoring provides a streaming bus that allows external tools to subscribe to specific topics to gather security events detected in the SIS. To collect the event messages it is required to implement a Kafka Consumer. There are multiple alternatives to implement Kafka clients, as it can be seen here: https://cwiki.apache.org/confluence/display/KAFKA/Clients. The authentication in this bus is implemented by digital certificate.

| Method | Resource (topic) | Content-type | Description |
|---|---|---|---|
| **SUBSCRIBE** | /nids_events_enact | JSON | Network Intrusion Detection System's security events. The format of these type of events are described here: https://suricata.readthedocs.io/en/suricata-4.1.4/output/eve/eve-json-format.html |
| **SUBSCRIBE** | /security_anomalies_enact | JSON | Anomaly events detected by the Security and Privacy Monitoring tool. |

In the following we provide details about the interface exposed by the SMOOL IoT Platform that enables the integration of security policies at design time by other tools. SMOOL has enriched it set of security features by adding built-in policy checks when creating KPs (SMOOL applications or clients). This security policy feature allows to control security aspects from its core, so the KP developer does not need to implement or wire the security flow into the business logic. Thanks to GeneSIS, the security features can be declared at design time, and when deploying and building the application, these features are translated into SMOOL security policies. GeneSIS can also deploy newer versions of the same application with more refined security controls by acting on the same SMOOL security built-in features. Therefore, either by using bare policies or enhanced policies, the SMOOL core will be able to accept or reject an incoming message, and this process in completely transparent to the developer of the KP.

| Method | Resource (topic) | Content-type | Description |
|---|---|---|---|
| **Java Class** | **org.smool.security.SecurityChecker#test(`AbstractOntConcept message`)** | N/A | This class verify if any message received fulfils the security policies. For ENACT, actuation messages should have valid authorization tokens, otherwise the message is rejected. This SecurityChecker could also be deployed from GeneSIS with more advanced features, like connecting to external security services or performing more specialized security constraints. More information can be found here: https://gitlab.com/enact/smool_enact/-/blob/master/CHANGES_V3.md |

### 2.3.10. Root Cause Analysis

In the following we provide details about the API exposed by RCA that enable the integration of RCA with other tools (from ENACT or not). A set of high-level commands are exposed by RCA in the form of a REST API.

| Method | Resource | Content-type | Description |
|---|---|---|---|
| **GET** | /rca/getall | Response: application/json | To retrieve all the records (known incidents) stored in the historical database |
| **GET** | /rca/getone | Response: application/json | To retrieve a record (a known incident) stored in the historical |

| | | | |
|---|---|---|---|
| | | | database identified by its ID |
| **POST** | /rca/insertone | Response: application/json Parameter: application/json | To report a newly detected incident with all the related traces so that it could be inserted to the historical database |
| **GET** | /rca/logs | Response: text/plain | To retrieve all the logs of the tool |
| **GET** | /rca/status | Response: application/json | To retrieve the current status of the monitored system and how it looks similar (in percentage) to the known incidents |
| **GET** | /rca/help | Response: application/json | To retrieve the API documentation |
| **POST** | /rca/update | Response: application/json Parameter: application/json | To push an update regarding a record (a known incident) in the historical database |
| **POST** | /rca/deleteone | Response: application/json Parameter: application/json | To delete a record (a known incident) in the historical database |
| **POST** | /rca/deletemany | Response: application/json Parameter: application/json | To delete multiple records (known incidents) in the historical database |

In the following we details the interaction between RCA and other ENACT tools.

#### *2.3.10.1. Root Cause Analysis as input provider for Online Learning*

The online learning tool needs information about the action space beforehand. Based on the size of the action space the actions coming from the algorithm needs to be clipped to fit for the current system to adapt. These information needs to be provided either by GeneSIS or Actuation Conflict Management, depending on which tool is responsible for the actual execution of an action. Additionally, RCA might pinpoint certain parameter values as the reason for a system to deviate from expected quality. In this case the action space used by the OL should be adapted and a new policy should be learned based on the new parameter boundaries. This manual adaptation of the action space might increase convergence. On the other hand, based on the computed reward the Online Learning tool should be able to shift the action selection probability so that it focusses on parameter values improving the quality (e.g., performance) of the system.

| | |
|---|---|
| Data to be exchanged: | Action space boundaries (i.e. dimensions of the parameter to be learned (e.g., 0.4-1.0 as values below 0.4 have been identified as a reason for the system to deviate from expected quality)) |
| Format: | JSON |
| Communication protocol: | Manual and/or REST over HTTP |
| Rationale | Online Learning proposes new parameter values for the software systems whose adaptation logic should be optimized. If certain ranges for the value can be excluded because they have been identified as a reason for the system to deviate from expected quality, this might have a positive impact on the learning process. |

#### 2.3.10.2. Root Cause Analysis as input provider for GeneSIS

In order to perform a root cause analysis, the Root Cause Analysis enabler needs to gather data from the SIS and in particular from its performances. RCA monitoring probes are thus required to be deployed together with the SIS. Whilst they are provided by the RCA enabler, they are deployed by GeneSIS.

| | |
|---|---|
| Data to be exchanged: | • Deployment model: a deployment model including the necessary probes to monitor the SIS's.<br>• RCA Monitoring probes: the implementation of the RCA monitoring probes. |
| Format: | • Deployment model: JSON<br>• Monitoring Probes: binary, ThingML code, etc. |
| Communication protocol: | Manual |
| Rationale | For GeneSIS to properly deploy a new version of the SIS including the RCA monitoring probes, it needs information about (i) where they should be located in the overall architecture of the system (i.e., where it should be deployed and its interactions with the rest of the system) and (ii) their actual implementation. |

# 3. Selecting the ENACT enablers for your needs

In order to accommodate and ease adoption to potential beneficiaries of ENACT, a set of questions were listed in the Table 4., It guides the users though the set of questions starting from Level 1 to Level 3. Based on the answers, the user is pointed to the exact enablers that best fit their needs. The questions can be cycled though multiple times in order to assess all the routes of outcomes.

## 3.2. Tools matching questions and decisions

Within the Table 4 a list of questions is presented along with the pointers to the tools which do respond to the requirements. Each of the positive possible answers leads either to the final answer pointing out the matching enablers or to the next level of Questions, as presented in the table.

| Question Level 1 | Outcome | Questions Level 2 | Outcome | Question Level 3 | Outcome |
|---|---|---|---|---|---|
| **Question about the type of users?** | Developing and maintaining IoT software | lead to second question on Level 1 | | | |
| | Integrator of IoT ecosystem | lead to third question on Level 1 but only with stage monitor, test, and plan | | | |
| **Questions about the type of DevOps activity?** | Code | Is your system running on heterogeneous platforms and hardware? | If Yes then check to ThingML enabler | | |
| | | Do you need support for integrating security mechanisms in your code? | If Yes then check to S&P Control enabler | | |
| | | Is your IoT system controlling actuators? | If Yes then check to ACM & OLE | | |
| | Test | Are you able to inject data into your IoT system? | If yes, read questions of level 3 | Would you be interested in understanding how your system react to real life data (including how your system is scalable)? | If yes, then check T&S |
| | | | | Test coverage on failures and security attacks is an issue for your team due to large amount and variety of data | If yes, then check T&S |
| | Deploy | Are you managing a fleet | If yes, then check DivEnact | | |
| | | Do you need to deploy software on devices with no direct connection to Internet | If yes, check GeneSIS | | |
| | | Do you need high availability deployment techniques | If yes, check GeneSIS | | |
| | | Do you need to deploy security mechanisms | If yes, check GeneSIS | Are you using an IoT middleware such as SOFIA | If Yes check GeneSIS, S&P, and SMOOL |
| | Operate | Do you need Access Control? | If yes, read questions of level 3 | Should you access control rules evolve in different contexts? | If Yes check CAAC; If no check S&P and CAAC |
| | | Do you want to make your system more resilient in the face of dynamic context? | Yes / no | Do you need support for actuator control strategies? | If Yes check OLE |

| | | | | | |
|---|---|---|---|---|---|
| | Monitor | Is your system in open context or several instances of your system may run in different contexts? | If yes, check RCA | Does your system involve actuators? | If yes, check BDA |
| | | Are you monitoring security of your system? | If no, check S&P | | |
| | Plan | Do you need to assess risk? | If yes, read questions of level 3 | Do you have requirements for compliance reporting? | If yes, check RM |
| | | Are you subject for regulatory compliance? | if yes, check RM | | |
| **Are you looking for security support?** | Yes / no | Do you need support for deploying your security mechanisms | If yes, check S&P | | |
| | | Do you need support for monitoring security | If yes, check S&P | | |
| | | Do you need support for Access control | If yes, check CAAC | | |
| | | Do you need support for integrating security mechanisms into your system? | If yes, check S&P | | |

*Table 4. List of questions to the users of ENACT framework.*

## 3.3. ENACT Tool Wizard

The same list of questions presented in the Table 4 is also used in a more interactive form, showcased in the Figure 2, where the user is guided though an online quiz that askes only questions which do apply to the user and lists the results in an easy consumable form with pointers to the integration with other Enact tools, the advancements against the current state of the art etc. Exact list of possible results can be found within the Table 5. The tool is publicly available via link on the project website or under the link: https://forms.office.com/Pages/ResponsePage.aspx?id=FjfNS8jFrkePQ7jfdpl4U_L3JSYwk8VLiMQMHWTkaYlUQkdaQTVMS05WOUQxSlpJVExMSFRQQUE1QS4u

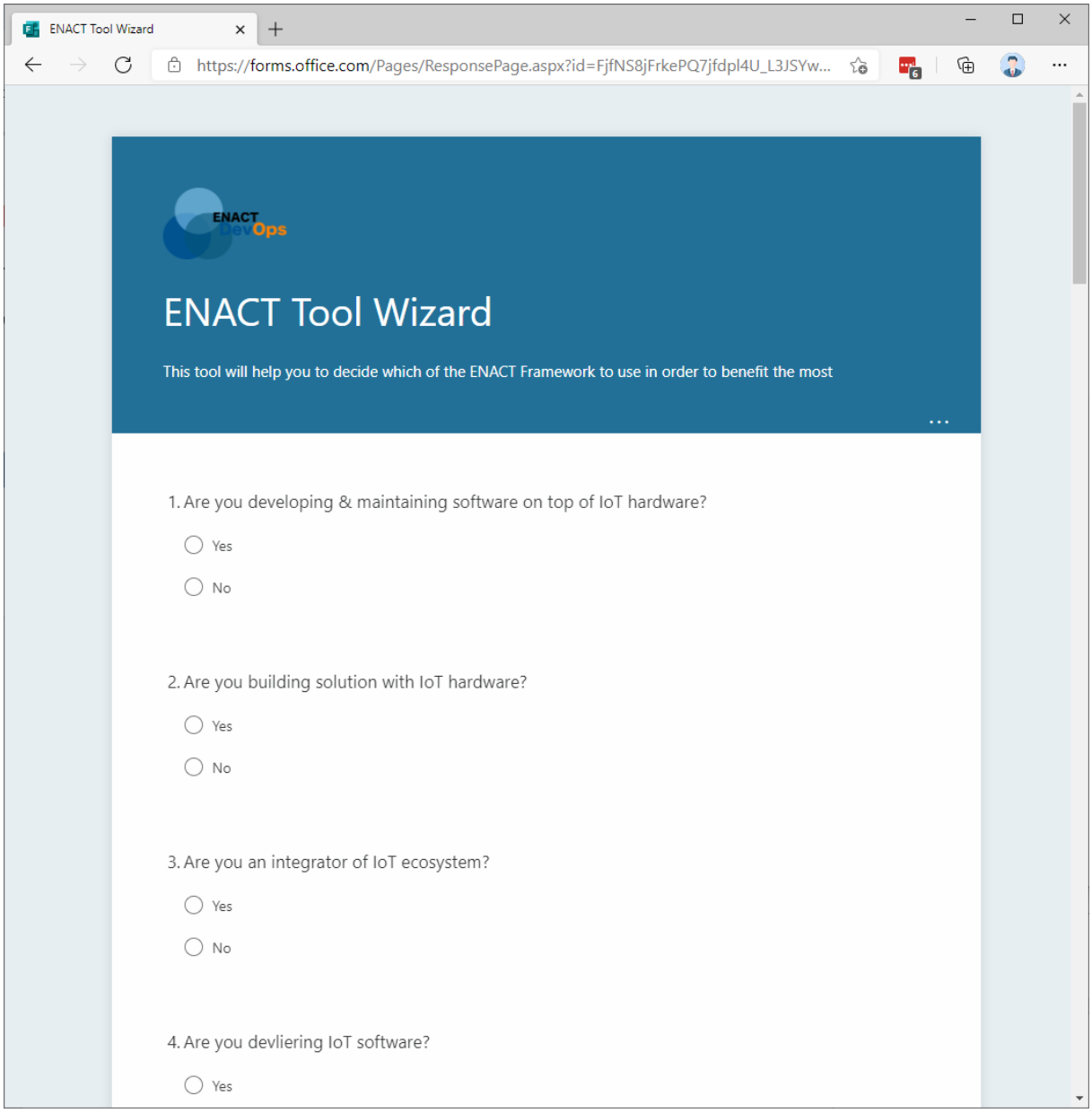


*Figure 2. Screenshots of the ENACT Tool Wizard*

| Stage | Enabler | Uniqueness & Benefit of the enabler compared to SOTA | Often bundled with? | Target user group | Cases where the enabler cannot be used |
|---|---|---|---|---|---|
| Code | ThingML | * Handle heterogeneity<br>* Off the shelf support for communication mechanisms<br>* Platform independent debugging, including on tiny devices | **ThingML + Diversifier:** Diversifier can be used to generated functionally equivalent versions of the SIS when using ThingML. In particular communication protocols can be diversified. **ThingML + S&P:** integration of security policies in IoT middleware on heterogeneous platforms **ThingML + GeneSIS:** automatic building and deployment of ThingML programs for the target platform, migration of software on heterogeneous software | Developers of SIS | |
| | S&P Control | Security policies definition in IoT Middleware | **S&P Control + GeneSIS:** the security features can be declared at design time, and when deploying and building the application, these features are translated into SMOOL security policies. | | Need IoT Middleware |
| | ACM | Identify and solve AC, using components checked against logical and temporal properties | **ACM + GeneSIS:** For the automatic identification and integration of ACM in SIS. Allow full integration of ACM in DevOps pipeline and thus the continuous actuation conflict management. | | Need Actuators |
| Test | T&S | Complement classic testing solution (unit, regression, etc.) for runtime or simulation based testing. Provide mean to explore tests situation that could not be easily anticipated via recording or failure, attack generation. Test based on real data - i.e. context-based testing | Interesting Provide hooks for being integrated with any Dev tool. Typically leads to a new Dev Cycle. | Developers of SIS and integrators | Being able to inject data in the SIS |

| | | | | | |
|---|---|---|---|---|---|
| **Release and deploy** | DivEnact | Management deployment over a fleet of devices with automatic assignment of software to large number of devices according to their contexts. | **GeneSIS + DivEnact:** GeneSIS for deployment of local subsystems and DivEnact for the fleet level | Developers, maintainers, and integrators | |
| | GeneSIS | The GeneSIS language includes security mechanisms as the first-class modelling elements<br>deployment delegation to support the devices with constrained resources or connectivity.<br>declarative deployment with high availability and the monitoring of deployment process. | GeneSIS + ThingML (cf. above)<br>**Delivery bundle:** GeneSIS + S&P + ThingML: automatic injection of security policies for software integrated with IoT platforms<br>GeneSIS + ACM (see above) | Developers, maintainers, and integrators | Need access to target device direct or indirectly via another device |
| **Operate** | CAAC | OAuth 2.0 standard protocol to make the provided authorizations responsive to the context, injecting contextual risk levels as dynamic attributes in the authorization mechanisms. | Provides hooks and API for integration in any software IT <-> OT | Developers, maintainers, and integrators | |
| | OLE | Use Reinforcement Learning algorithms via Web API to learn actuation control strategies and supervise learning progress with customizable monitoring capabilities. | **BDA + OLE:** Additional metric to guide learning process | | |
| | BDA | Monitoring and analysing effectiveness of SIS (their effect on environment) analysing symptoms of drifts in effectiveness of SIS from the systemic model perspective | **BDA + RCA:** BDA provides behavioural drifts alerts and their associated symptoms to help RCA to imprice the historical data<br>**BDA + OLE:** use behavioural drift value as a reward for the reinforcement learning approach (OLE could learn on a higher level)<br>In general all monitoring tools are interesting to be coupled with Dev Tools as a mean for feedback | Maintainers and integrators | Need external observation data, so the corresponding sensors on the fields |

| | | | | | |
|---|---|---|---|---|---|
| **Monitor** | S&P | Continuous monitoring of the status of the system security and privacy. It uses machine learning to correlate data captured by multiple distributed monitoring agents deployed in different layers, in order to offer a holistic view of the SIS and enable the detection of advanced attacks. It is fully elastic for the rapid scaling of the target systems. | **S&P Control:** Security aware SMOOL clients work as a monitoring agent for S&P and provide security events related to IoT platform and its communications. Moreover, S&P control can act against potential attacks by requesting SMOOL Server to block the potential.<br>In general all monitoring tools are interesting to be coupled with Dev Tools as a mean for feedback | Maintainers and integrators | |
| | RCA | Can be used in different context, learns errors and compares the similarity between the observed symptoms with the recorded ones to identify the possible incidents with the corresponding root-causes, impacts and mitigation actions | RCA helps systemizing the experience dealing with faults, failures and problems as well as to avoid and react more quickly against their occurrences. | Maintainers and integrators | |
| **Plan** | RM | Can seamlessly integrate with DevOps cycle though deep two ways integration with Jira and Git. Provides means to detect, understand and act upon risk as well as monitor treatment implementation and effectiveness at the later stage of the project. | **Genesis:** provides base understanding of the architecture at hand | Developers of SIS and integrators | |

*Table 5. List of possible answers with value proposition of each of the Enabler.*

Within next section we describe how the ENACT enablers contribute to trustworthiness aspects and how they benefit potential users of the enablers.

# 4. Summary of enabler's contribution to trustworthiness

In this section we first summarize the contribution of the enabler in terms of supporting the development and operation of trustworthy SIS. Secondly, we report on the trustworthiness requirements from the use cases towards the enablers.

Based on the NIST's definition of trustworthiness for Cyber Physical Systems[9], within ENACT, we adopt the following definition of trustworthiness and its different properties, within ENACT:

"**Trustworthiness** refers to the preservation of security, privacy, safety, reliability, and resilience of SIS".

We adopt the following definitions of the different properties:

- **Security** refers to the preservation of confidentiality, integrity and availability of information[10].
  - **Integrity** is the property of protecting the accuracy and completeness of information[11].
  - **Confidentiality** is the property that information is not made available or disclosed to unauthorized individuals, entities, or processes[3].
  - **Availability** is the property of information being accessible and usable upon demand by an authorized entity[3].
- **Privacy** refers to the protection of personally identifiable information (PII)[12]. 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.
- **Safety** refers to the ability of the cyber-physical system (CPS) to ensure the absence of catastrophic consequences on the life, health, property, or data of CPS stakeholders and the physical environment[1].
- **Reliability** refers to the ability of the CPS to deliver stable and predictable performance in expected conditions[1].
- **Resilience** refers to the ability of the CPS to withstand instability, unexpected conditions, and gracefully return to predictable, but possibly degraded, performance[1].

The table below summarizes how each individual ENACT enabler contributes to the development and operation of trustworthiness of SIS. It is also important to note that the main promise of the ENACT project in order contribute to trustworthiness is to allow DevOps for SIS. This is because the support offered by the ENACT enablers to the DevOps of SIS is, by itself, a major contribution for supporting the trustworthiness aspect. Indeed, the adoption of the DevOps principles and practices in the field of the IoT is decisive to enable the continuous and agile evolution of SIS, which is necessary to adapt the system to newly appearing trustworthiness threats and to ensure its overall quality.

[9] Edward R. Griffor, C.G., David A. Wollman, Martin J. Burns, *Framework for Cyber-Physical Systems: Volume 1, Overview*. June 26, 2017, National Institute of Standards and Technology.
[10] ISO/IEC, 27000: 2012, Information technology - Security techniques - Information security management systems - Overview and vocabulary. 2012, International Organization for Standardization.
[11] Bishop, M., *Computer security: art and science*. 2003: Addison-Wesley Professional.
[12] .ISO/IEC, 29100.2011 Information technology—Security techniques—Privacy framework. 2011, International Organization for Standardization.

| Enabler | Security | Privacy | Reliability | Resilience | Safety |
|---|---|---|---|---|---|
| **ACM** | No. | No. | **Yes.** To ensure the reliability of the system embedding actuation conflict managers, they are checked against logical and temporal properties. In addition, it helps solving the case when a SIS does not behave as expected due to indirect actuation conflict. | No. | **Yes.** The management of actuation conflicts can have a direct impact on the safety of the system (i.e., unmanaged conflicts can lead to safety issues) |
| **BDA** | No. | No. | **No**. Because drift should appear only under unexpected conditions, does not focus on performances. | **Yes.** By monitoring and analysing drifts to trigger a new development cycle. | No. |
| **CAAC** | **Yes.** Confidentiality: ensure proper access control to the data. | **Yes.** Ensure privacy in the data managed by SIS, since the control over the personal data is kept by their owner | **Yes.** Information is made available or disclosed only to authorized individuals, entities, or processes, in a controlled way. | No. | No. |
| **DivEnact** | **Yes.** By diversifying software at the Edge, we implement a "moving target" defence strategy. | No. | **Yes.** Diversifying software can lead to an increase in their reliability. | **Yes.** Support failure recovery in a DevOps process. | No. |
| **GeneSIS** | **Yes.** Support specification of security requirements and the deployment of security mechanisms. | **Yes.** Indirectly via security. Can be used to deploy privacy mechanisms | **Yes.** Via mechanisms such as blue/green deployment. | **No.** Not directly, only by supporting adaptation. | No. |
| **Online Learning (OLE)** | No. | No. | **Partially.** OLE focuses at system operations under aspects unexpected conditions. However, the explainable AI element of OLE facilitates that online learning itself happens in a reliable fashion. | **Yes.** Adaptation and improvement of adaptation logic to keep working in unseen environments. | No. |
| **RCA** | **Yes.** To detect the cause of security flaws | **Yes.** Indirectly via security | **Yes.** Root cause of previously known performance problems | **Yes.** Root cause of previously unknown performance problems supported by ML techniques | No. |
| **Risk Manage** | **Yes.** Confidentiality, Integrity and Availability | **Yes.** Predictability, Manageability, | **Yes.** It can be provided depending on the vulnerabilities defined, risks | **Yes.** It can be provided depending on the vulnerabilities | **Yes.** It will be taken into consideration |

| | | | | | |
|---|---|---|---|---|---|
| **ment Enabler** | can be provided depending on the vulnerabilities defined, risks identified, and mitigation actions proposed by the user or our knowledge bases | Disassociability can be provided depending on the vulnerabilities defined, risks identified, and mitigation actions proposed by the user or our knowledge bases | identified, and mitigation actions proposed by the user or our knowledge bases | defined, risks identified, and mitigation actions proposed by the user or our knowledge bases | through the analysis of the impact of risks. |
| **S&P Monitoring and control** | **Yes.** Monitors and controls confidentiality, integrity and availability of data | **Yes.** Monitors and controls confidentiality, integrity and availability of personal data | **Yes.** Monitors availability | No. | No. |
| **Test & Simulation** | **Yes.** Test & simulate the system under cyber-attacks situation to verify the security of the system | No. | **Yes.** Test & simulate the system in expected situations to verify the reliability of the system. | **Yes.** Test & simulate the system in unexpected situations to verify the resilience of the system | No. |

*Table 6. Enabler's contribution to trustworthiness*

Some enablers are marked as indirectly contributing to the privacy property. This is because we believe the support for security provided by these enablers also contributes preserving the privacy of a SIS. The same applies to the safety property, the contributions of the enablers on security, privacy, reliability and resilience properties are important to help ensuring the safety of a SIS.

Several of the enablers from the monitoring and analytics layer of the ENACT DevOps Framework (i.e., security and privacy monitoring, behavioural drift analysis, and root cause analysis) are considering security, privacy, reliability and resilience aspects. It is worth noting that these tools are complementary and not overlapping: On the one hand, the security and privacy monitoring enabler focuses on observing **symptoms** of security and privacy issues, and the behavioural drift analysis enabler focus on **symptoms** of reliability and resiliency issues. On the other hand, the root cause analysis focuses on understanding the **causes** of these symptoms.

Within the next section we showcase the achieved integration of the ENACT enablers with widely adopted DevOps or IoT tools. By enabling such integration we foresee greater potential for user adoption to carry the effort of the project beyond ENACT.

Copyright © 2018-2020 by the ENACT consortium – All rights reserved.

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

# 5. Enablers interoperability with market leaders

| Enabler | Integrate with | How | License of the framework it is integrated with | Contribution to the integrated platform/framework/tool if any | Status |
|---|---|---|---|---|---|
| **GeneSIS** | Node-RED | GeneSIS support deployment and configuration of Node-RED containers | Apache v2 | Set of nodes used for the deployment agents - i.e; deployment and compilation of ThingML programs, deployment of Arduino sketches, etc. | Done |
| **GeneSIS** | Ansible | GeneSIS can use Ansible as one of the GeneSIS deployment mechanisms | GPL 3 | N/A | Done |
| **GeneSIS** | ThingML | GeneSIS automatically compiles and deploys ThingML programs, leveraging the ThingML CLI | Apache v2 | GeneSIS can automatically the logging mechanism enabling ThingML program remote debugging, until now this was done manually | Done |
| **GeneSIS** | SMOOL | Deployment of SMOOL security KP (see D2.3 for more details) | Apache v2 | GeneSIS can automatically inject security policy into a SMOOL KP before automatically buiding and deploying it | Done |

Copyright © 2018-2020 by the ENACT consortium – All rights reserved.

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

| | | | | | |
|---|---|---|---|---|---|
| **GeneSIS** | FIWARE | Automatic deployment of Fiware orion context broker (see D2.2 for more details) | AGPL 3 | N/A | Done |
| **GeneSIS** | Jenkins | Jenkins can trigger a deployment using the GeneSIS API (e.g., once some software is built) | MIT | N/A | Done |
| **ThingML** | SMOOL | Automatically generate ThingML code that integrate with SMOOL KP (see D2.3 and D4.3 for more details) | Apache v2 | Provide a uniform way to build SMOOL KPs | Done |
| **Actuation Conflict Management** | Node-RED | Detect and Manage Actuation Conflict in Node-Red applications | Apache v2 | N/A | Done |
| **Actuation Conflict Management** | OpenHab | Access to IoT hardware | Eclipse Public License 2.0 | N/A | Done |
| **Actuation Conflict Management** | ThingML | Detect and Manage Actuation Conflicts in ThingML programs | Apache v2 | N/A | Done |
| **Context Monitor and Behavioural Drift Analyzer** | Node-RED | Facilitate context monitoring and behavioural drift analysis | Apache v2 | N/A | Done |
| **Context Monitor and Behavioural Drift Analyzer** | SMOOL | Behavioural drift analysers from contextual information from SMOOL | Apache v2 | N/A | Done |

Copyright © 2018-2020 by the ENACT consortium – All rights reserved.

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

| | | | | | |
|---|---|---|---|---|---|
| **Context Monitor and Behavioural Drift Analyzer** | Home I/O | Simulate Smart Building | Commercial | Develop a MQTT publisher to easily access Home I/O simulation. Home behaviour can be easily monitored and analysed in the simulated environment. | Done |
| **Context Monitor and Behavioural Drift Analyzer** | OpenHab | Access to IoT hardware | Eclipse Public License 2.0 | N/A | Done |
| **Context Monitor and Behavioural Drift Analyzer** | OpenBVE | Train simulator | Public Domain | Develop a MQTT publisher to easily access OpenBVE data. Train behaviour can be easily monitored and analysed in the simulated environment. | Done |
| **Context Monitor and Behavioural Drift Analyzer** | Mosquitto | MQTT Broker | EPL 1.0 | N/A | Done |
| **Security control and monitoring enabler** | SMOOL | SMOOL client only for security and change in the SMOOL core ontology | Eclipse EPL V1.0 | Security control & monitoring | Done |
| **Security control and monitoring enabler** | Suricata | Suricata monitors other systems (e.g., LAN) and provides the metrics/ indicators as well as the alerts of anomalies | GPLv2 license | N/A | Done |

Copyright © 2018-2020 by the ENACT consortium – All rights reserved.

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

| | | | | | |
|---|---|---|---|---|---|
| **DivEnact** | Microsoft Azure IoT Hub | Use Microsoft IoT Hub for Edge device management | MIT License | N/A | Done |
| **Online Learning Enabler** | Mosquitto https://mosquitto.org/ | Broker for MQTT. | EPL 1.0 | N/A | Done |
| **Root Cause Analysis Enabler** | MMT | MMT monitors other systems (e.g., WSN, 5G) and provides the metrics/ indicators as well as the alerts of anomalies | Proprietary | N/A | Done |
| **Root Cause Analysis Enabler** | Suricata | Suricata monitors other systems (e.g., LAN) and provides the metrics/ indicators as well as the alerts of anomalies | GPLv2 license | N/A | Done |
| **Context-aware Access Control** | WAM | Context-aware Access Control for IoT - REST API | Proprietary | Evolution of the authentication and authorization mechanisms provided by Evidian Web Access Manager (WAM) intended for the Internet of Things. Provides access control for both users and devices. | Done |
| **Context-aware Access Control** | TellU Cloud Gateway | To provide state of the art integration to the TellU Cloud | Proprietary | N/A | Done |

Copyright © 2018-2020 by the ENACT consortium – All rights reserved.

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

| | | | | | |
|---|---|---|---|---|---|
| **Risk Management Enabler** | PDP4E Risk Management tools | Open source outcomes from PDP4E generated in parallel to ENACT continuously integrated | EPL v2.0 | Contribution to PDP4E by helping them to strengthen continuous risk management capabilities | Done |
| **Risk Management Enabler** | JIRA | Inject features in the backlog | Proprietary | N/A | Done |
| **Risk Management Enabler** | Git | Collect evidences on the status of the features as they are developed | GNU GPLv2 | N/A | Done |
| **Risk Management Enabler** | CWE | Collect Vulnerabilities and Threads from the catalogue | N/A | N/A | Done |
| **Risk Management Enabler** | Open ID (AD Office 365) | Integrate with active directory for Identity control | Commercial | N/A | Done |
| **Test & Simulation** | Node-RED | Visualize and test the Output and the Input of the simulation | Apache v2 | N/A | Done |
| **Test & Simulation** | Mosquitto | Test and Simulation can communicate with Mosquitto via MQTT protocol | EPL/EDL licensed | N/A | Done |
| **Test & Simulation** | ActiveMQ broker | Test and Simulation can communicate with ActiveMQ broker via STOMP protocol | Apache v2 | N/A | Done |

Copyright © 2018-2020 by the ENACT consortium – All rights reserved.

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