Title: *Agile Integration Methodologies and Tools*

Authors: *Franck Dechavanne (CNRS), Nicolas Ferry (SINTEF), Anne Gallon (EVIDIAN), Alberto Huélamo (CA), Stéphane Lavirotte (CNRS), Jean-Yves Tigli (CNRS), Hui Song (SINTEF).*

Editor: *Jacek Dominiak (CA)*

Reviewers: *Anne Gallon (EVIDIAN), Modris Greitans (EDI)*

Identifier: *Deliverable # D5.1*

Nature: *Report*

Date: *October 2019*

Status: *Final*

Diss. level: *Public*

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

Executive Summary

The primary scope of the deliverable is to describe the methods and tools used in order to achieve successful project outcomes from the technical standpoint. It showcases the agreed methodologies and tools that will be used throughout the project delivery. It also describes in detail the reasoning behind the methodological choices for the project and the novel approach to the European project delivery. Secondly, the deliverable describes the first approach to the formation of the ENACT framework and explains different levels of the structure of the architecture and the logical split between the different levels.
Lastly, the deliverables lists all the enablers which are being developed within the project and provides the insights into the planned functionality, actors to whom the enablers are designed, and provides the reasoning behind the choice of functions.

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

Members of the ENACT consortium:

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

Revision history

Date	Version	Author	Comments
18/09	V0.1	Jacek Dominiak (CA)	Table of content
25/09	V0.1	Nicolas Ferry, Hui Song (SINTEF)	Sections 3.1.4, 3.1.6
26/09	V0.1	Anne Gallon (EVIDIAN)	Section 3.1.7
02/10	V0.1	Franck Dechavanne, Stéphane Lavirotte, Jean Yves Tigli (CNRS)	Sections 3.1.5, 3.1.12
02/10	V0.1	Nicolas Ferry (SINTEF)	Description architecture
15/10	V1	Jacek Dominiak (CA)	Fist version sent to review
30/10	V1.1	Jacek Dominiak (CA)	Addressed reviewers' comments
31/10	V1.1	Jacek Dominiak (CA)	Release

Contents

1 Introduction

1.1 Context

The objective of this deliverable is to introduce the set of agreements which has been done within the project consortium in order to enhance the typical delivery of the usual waterfall-based structure of EU project. Throughout ENACT, the consortium is committed to deliver best quality results possible, not only from the academical and research standpoint, pushing the boundaries of what is currently possible in IoT space, but also in the technical deliveries of the project. This can be in part achieved by set structured exercises, similar to these, typical DevOps teams have to go through. The objective of the project is to build the toolset for the IoT DevOps teams using and in part enhancing the current understanding of what DevOps means.

By introducing the lean and agile aspects to the project structure we aim to showcase how trustworthiness can be enhanced by proactive and adaptable tooling and methods used against the often approach of only analysis.

Deliverable D5.1 also aims to introduce the idea of progressive exploitation of the project results by starting the exploitation activities early in the project, as early as the M12. This approach certainly helps to mitigate the potential lack of timeliness and the adaptability of the project results throughout taking into account the ever-changing aspect of technology against the project duration.

Finally, it is worth noting that this deliverable is aiming to provide a wraparound D2.1, D3.1 and D4.1 by introducing the overall concept of the ENACT architecture and by showcasing which tools and methods will be used to develop the proposed enablers.

To conclude, Section 2 aims to provide an overall vision on what is agile research and how it should be conducted. Section 3 showcases the enablers and their scope and Section 4 talks about the integration methods and tools which will be used during the project.

1.2 Achievements

Objectives	Achievements
Setting of methods and tools used for the project integration.	We conducted an extensive tools analysis and cherry picked the ones which not only did meet the requirements but also provided additional technical values.
Initial concept of the overall ENACT architecture	From the analysis of the requirements of the use cases, as well as the concepts of the tools described in deliverables D2.1, D3.1 and D4.1 we derived initial set of ENACT architecture.
Integration plan	The plan was agreed to produce the enablers mock-ups early in the project. These mock-ups will allow the tool developers to understand what the features are needed for the enablers. They will

	also be used as dissemination and exploitation materials. Set of technological standards was agreed in order to guarantee coherence of the project results.
Requirements elicitation from the perspective of the user of the ENACT enablers	Based on D1.1 describing the use cases requirements as well as previous work of parties involved, set of user stories were created in order to give a sense of enablers scope.

2 Agile Approach in ENACT

2.1 Agile methodologies in research projects

The world and the market are constantly changing, creating a lot of uncertainty and opportunities for business. In the current application economy and with emerging technologies that enable systems that expand across the cloud, IoT and the edge, companies need to find ways to adapt rapidly and cost efficiently to changes in the conditions and behaviour of the customers. The market demands agility in the product/service development process. Flexibility and continuous alignment with customer needs are also key to reach the market faster and more efficiently.

Organizations in all industries recognize they are continuing their evolution into technology and data companies[1], and their business models are being partially or fully transformed by software. As businesses in many industries become software-driven, the need for innovation forces companies to replace traditional software development methods such as the waterfall, in favour of software development, described by Royce[2], or similar alternatives. These alternatives must guarantee continuous development and delivery models in an aggressively changing market. A popular solution for many companies is the adoption of agile methodologies[3].

At the same time, innovation capacity becomes more and more important. However, although research and innovation bring clear value to software industry, it is not straightforward to align research outcomes with corporate strategy and agile software development. This may jeopardize the potential impact and business value generated by research and generates obvious challenges for industry when structuring research to guarantee impact. Because of agile methodologies, software project's value is usually specified based on a short-term vision in the early planning phase, and it keeps continuously evolving over the project lifetime depending on changing customers' requirements. Organizations expect high value from their research and innovation processes and they set up different strategies and mechanisms to transfer research outcomes into the business. However, in most cases, these strategies do not take into consideration dynamic business environments and generate little impact.

Continuous innovation is a sustainable process that is responsive to evolving market conditions and based on appropriate metrics across the entire lifecycle of planning, development and run-time operations[4]. However, when multiple stakeholders are involved in delivering and capturing project value (say for instance researchers and business unit product owners), their different viewpoints need to be

[1] W. W. Royce. Managing the development of large software systems: Concepts and techniques. In Proceedings of the 9th International Conference on Software Engineering , ICSE '87, pages 328–338, Los Alamitos, CA, USA, 1987. IEEE Computer Society Press.
[2] McKendrick, J. (2015, April 30). Every Company Now A Technology Company: Latest Round Of Mergers And Acquisitions Confirms It. Forbes. Retrieved from http://tinyurl.com/j6f7ub5
[3] Narendra Kurapati, Venkata Sarath Chandra Manyam, and Kai Petersen. Agile Software Development Practice Adoption Survey , pages 16–30. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.
[4] Cole, Robert E. "From continuous improvement to continuous innovation." Quality Management Journal 8.4 (2001): 7-21

considered at the beginning of the project[5], and continuously renegotiated during the project lifetime[6]. Integrating different perspectives from different stakeholders in a continuous innovation framework is an area of active research. In general, consciously or unconsciously, organizations that fund research still expect a solid, waterfall-based linear research plan[7] that does not usually match with the incremental and iterative approach followed by software companies. Jarvinen et al. present the Lean Research Approach for Industry-driven Research (LRA)[8] to facilitate collaborative research between several organizations that have mutual interests in rapidly changing environments. Research goals are defined in a roadmap for the research work called Strategic Research and Innovation Agenda (SRIA)[9]. Their proposal is based on two key elements: Continuous Planning and the Research Sprint model.

2.2 Adoption of Agile methodologies in ENACT

Research projects just by their extent complexity make it challenging for the software development agile methods to be directly applicable and exploitable. Scope, framework and multitude of parties involved of EU funded project prone it to be even more challenging than typical R&D exercise. EU projects are required to plan their run into what appears to be the typical waterfall delivery fashion. The project planning is greatly executed prior to the project even being submitted, with, if granted, the commitment of the direction for the next 24-48 months. This does not take into consideration the uncertainty of the future and leaves little space to the adaptation, or even possible pivot, of the project outcomes to the ever-changing landscape of technology.

Within ENACT, regardless of the structure proposed, we believe that we can benefit from using some of the agile methodologies to enhance the project outcomes. We believe that these agile tools and methods methodologies will help not only to enhance our technical output of the project but also enhance the project exploitation in the future. The list bellow showcases all the tools and methods ENACT adopted along with the reasoning. These are:

1. **MVV identification (Minimum Viable Value)**: this agile exercise is trying to determine the minimum set of steps the actor needs to perform in order for the tool to provide the benefit. ENACT will aim to have its enablers set within the score of 5.
2. **User Stories**: requirements for the user actions and the tools feature sets, along with the actors and the reasoning of the usefulness of the features are represented in the form of user stories[10]. This aims to give at a glance view of which actors' actions will be required within the enabler as well as what is the value it provides.

[5] Bjørn Johs Kolltveit and Kjell Grønhaug. The importance of the early phase: the case of construction and building projects. International Journal of Project Management, 22(7):545–551, 2004.
[6] Wijnand Veeneman, Willemijn Dicke, and Mark De Bruijne. From clouds to hailstorms: a policy and administrative science perspective on safeguarding public values in networked infrastructures. International journal of public policy , 4(5):414–434, 2009.
[7] Janne Järvinen, Tua Huomo, and Tommi Mikkonen. Running software research programs: An agile approach. In Proceedings of the 39th International Conference on Software Engineering Companion , ICSE-C '17, pages 314–316, Piscataway, NJ, USA, 2017. IEEE Press.
[8] Janne Järvinen, Tua Huomo, and Tommi Mikkonen. Running software research programs: An agile approach. In Proceedings of the 39th International Conference on Software Engineering Companion , ICSE-C '17, pages 314–316, Piscataway, NJ, USA, 2017. IEEE Press.
[9] P. Kettunen, P. Kuvaja, A. Koivisto, C. Lassenius, P. Lehtovuori, S. Lilja, S. Miettinen, T. Mikkonen, J. Munch, T. Mannisto, M. Oivo, J. Partanen, I. Porres, J. Still, T. Huomo, J. Jarvinen and P. Tyrvainen. Strategic research agenda for need for speed (n4s).http://n4s.fi/articles/SRIA_Need4Speed_V5_0_April_2015.pdf, 2015. Accessed: 2018-10-02.
[10] https://en.wikipedia.org/wiki/User_story

3. **Flow charts**: identification of the user actions will be represented in the form of flow-charts. This allows for greater understanding, identification and simplification of the user flow process through the tool in order to achieve lower mark of MVV.
4. **Mock-ups**: all the ENACT enablers which do require graphical use interface, will present the implementation of the steps described in the flow charts in the form of the tool mock-ups. This will be then used as a help and motor of the dissemination and exploitation actions of the ENACT framework or each of its components. ENACT is aiming to have all the mock-ups ready by the M12 of the project.
5. **Unified API definition**: to ease adoption of the project outcomes, a set of API standards is agreed. More detailed description of the API definitions used can be found under Section 4.2.
6. **Unified UI definition**: following the reasoning of the unified API definition, UI definition for all ENACT enablers has been decided and it is described in detail in Section 4.3 of the document.
7. **Continuous Integration**: enabled for all the open source enablers released during the project duration. The outcomes of the project will be integrated automatically and deployed in a truly agile fashion, where only if all the level of automatic testing is passed, the enabler will be deployed to a demo state.
   Continuous Deployment: enabled for all the open source enablers which will have a deployable state or will be packaged into a download. This action will be only triggered if the Continuous Integration steps is successful.
8. **Release back-compatibility from M15 of the project onwards**: following the initial release of the tools in the M15 of the project, ENACT will aim not to introduce breaking API changes and if ones are required, versioning of the API will be used instead.
9. **Pair programming and knowledge transfer**: set of online meetings are being scheduled throughout the project in order to level up the technical side of the project and allow all parties involved technical coherence in the results delivery. Exact set of sessions identified will be described in D5.2.

All the Agile methods and tools described above aim to ease the project outcome adoption and ensure that the efforts can be disseminated the earliest possible.

In the next chapter we will describe the overall ENACT architecture and introduce all the enablers with their core features, features reasoning and the user centric steps which are required in order to get the added value.

3 Integration planning

3.1 ENACT architecture

ENACT will provide an integrated DevOps Framework composed of a set of loosely coupled tools. Still, these tools can be seamlessly combined, and they can easily integrate with existing IoT platform services and enablers. Figure 1 shows the set of tools that forms the ENACT DevOps Framework as well as the relationships between these tools. This conceptual architecture consists of five layers, where each layer denotes a particular level of abstraction, complexity and dynamic. Note that these layers retain a certain degree of independence so that (i) the implementation of one layer can be easily substituted by another with minimal impact on the other layers, and (ii) each of them can pursue its own dynamic.

From the most abstract to the most concrete (i.e., from the farthest to the closest to the running system), the layers are described as follows:

1. **Improvement Layer**: This layer provides the mechanisms to continuously improve and manage the development and operation processes of trustworthy SIS. On the one hand, the Risk Management tool will help organizations analysing the architecture of their Smart IoT Systems and detecting potential vulnerabilities (in particular related to security and privacy aspects). In addition, it will be able to understand how vulnerabilities and potential associated risks may impact the software development process. On the other hand, the online learning tool will focus on improving the behaviour of the adaptation engine that will support the operation of trustworthy SIS. In general, the improvement layer provides feedback and knowledge to the adaptation layer with the aim to improve it.
2. **Adaptation Layer**: This layer first embeds a set of editors to specify the behaviour as well as the orchestration and deployment of SIS across IoT, Edge and Cloud infrastructure. These editors all integrate with mechanisms to maximize and control the trustworthiness of the system. The activities performed at this layer are strongly affected by the inputs from the improvement layer.
3. **Enactment Layer**: The aim of this layer is to actually enact the deployment and adaptation actions decided at the Adaptation Layer. The mechanisms of this layer monitor and manage the deployment of the running system.
4. **System Layer**: This layer consists of the running system together with the environment and infrastructure in which it executes. This includes both production and testing environments.
5. **Monitoring and analytics Layer**: This layer is orthogonal and feeds the other four. It provides mechanisms to monitor the status of the system and of its environment. This includes mechanisms to monitor the security and privacy of a smart IoT system. In addition, it performs analytics tasks providing: (i) high level notifications with insights on ongoing security issue, (ii) diagnostics and recommendations on system's failures, and (iii) feedback on the behavioural drift of smart IoT systems (i.e., system is functioning but not delivering the expected behaviour).

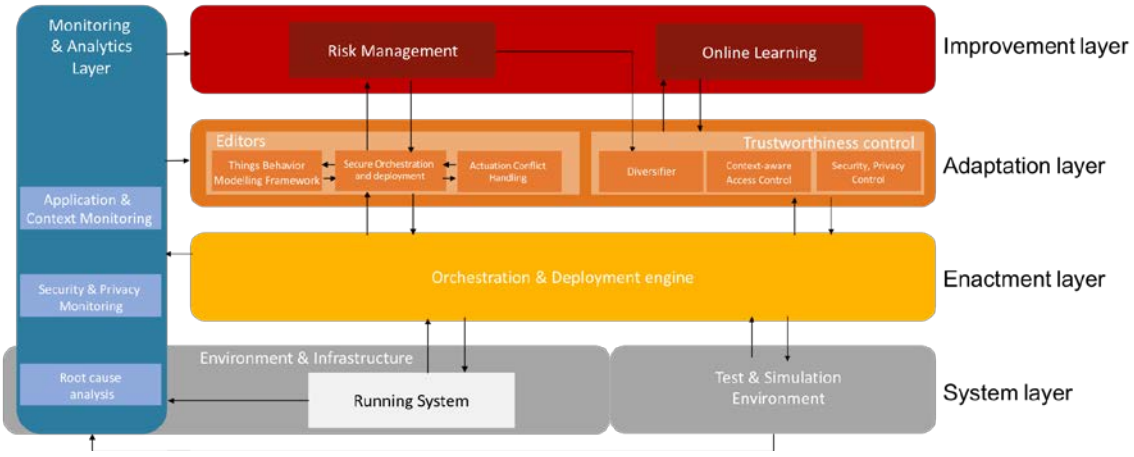

Figure 1. Overall Architecture of ENACT

As for the actors, given the fact that ENACT aims to support full DevOps cycle, it is foreseen to categorized the actors involved with the process in two main categories. These are:

- IoT DevOps engineer – this is a team member of a DevOps team, which handles the typical functionalities of a developer or is using DevOps tools in order to conclude operations part

in a programmatical manner. These actors include roles such as Developer, Software Architect, Scrum Master etc.

- System Operator – this is an actor which handles the ongoing operations of the system and the application and can provide the feedback to DevOps team to influence further developments or configuration specifications

In the following sections we detail all the tools that will form the ENACT DevOps Framework with their appropriate actors.

3.1.1 Risk Management

The Risk Management Enabler is a tool to help organizations to detect vulnerabilities in their IoT Systems' architecture and predict the impact on the software development process. On the one hand, this enabler will help organizations to analyse the architecture of their Smart IoT Systems and detect potential vulnerabilities, thus supporting the selection of the right components in the system by detecting characteristics that may be important to mitigate those vulnerabilities. On the other hand, it will be able to understand how vulnerabilities and potential associated risks may impact the software development process, for instance in terms of generating delays with respect to planned tasks and jeopardizing the ability of a company to continuous deliver according to the commitments with customers.

The user stories introduced in Listing 1 detail the features that should be offered by Risk Management.

Listing 1. Risk Management user stories

Actors

- IoT DevOps Engineer
- Risk Manager (change to SAFe based actors, possibly RTE [Release Train Engineer] AND/OR Product Owner)

User stories

- As an IoT DevOps Engineer, I want to know what my functional risks are, so that I th 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 managed.
    - 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.
    - 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 an IoT DevOps Engineer, 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 DevOps Engineer, I want to gain insights into the all types of risks, so I can influence the technical requirements of the software.
- As a Risk Manager I want to be able to reliably specify the risk likelihood and consequence based on agreed methodology so that I can evaluate the risks and prioritize the resolution.
- 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.
- As a Risk Manager I want to be able to know what the minimum viable risk level is where the "just enough" scenario of risk management is reached, so I can be aware what mitigation actions are strictly necessarily.
- As an IoT DevOps Engineer I want to be able to monitor the status of my risks, so I can be aware if some risks were reintroduced or not.
- 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
- As an IoT DevOps Engineer 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.
- As a IoT DevOps Engineer I want to detect weak points of the architecture and the associated risks, so that I can detect related risks.

In the figure bellow a flow of the user actions though the Risk Management enabler is presented. The step groups are introduced in columnar fashion. Only blue steps are required from the user to take the action upon.

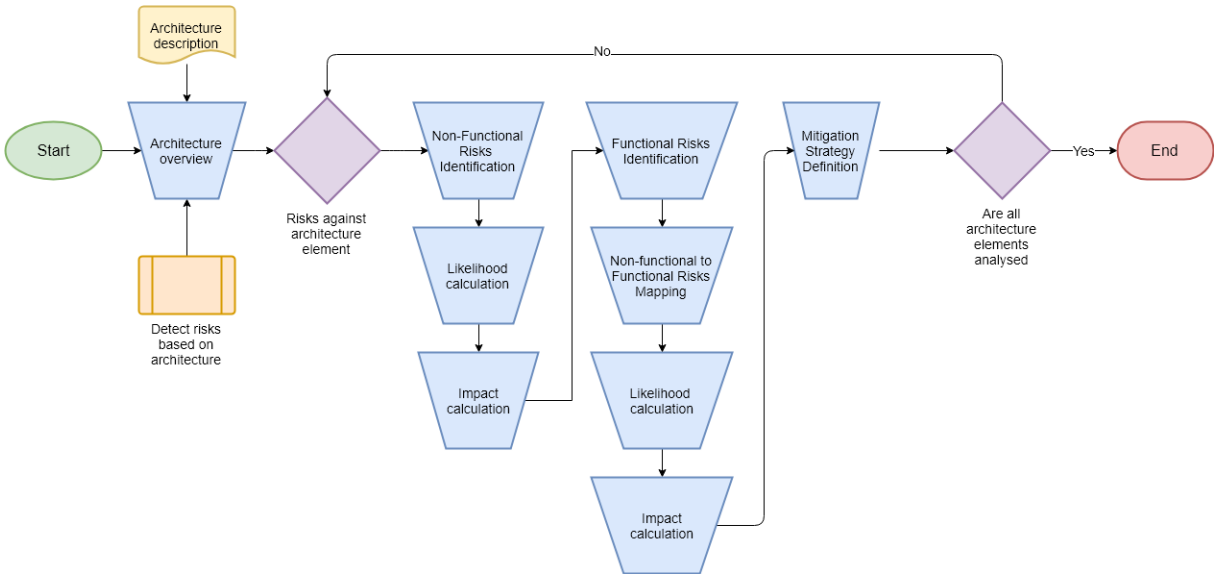

Figure 2. Flow-chart of the Risk Management user interface

3.1.2 Online Learning

The Online learning enabler offers a tool for the maintenance and continuous improvement/updating of an adaptation logic's knowledge base that contains adaptation rules (e.g. which system configurations

to be used in certain environment situations) for the self-adaptation of the system. Thereby, the adaptation rules are continuously improved during the operation of the IoT system.

The improvement is done according to the knowledge base of the online learning tool which is continuously updated by means of one or more learning techniques, in particular using Reinforcement Learning.

Listing 2. Online Learning user stories

- **Actors:**
  - IoT DevOps engineer

- **User Stories**
  - 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.
  - As an IoT DevOps engineer, I want the online learning enabler to perform automatically, so that no human intervention is needed.
  - 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.
  - 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.
  - 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.
  - 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.
  - 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.
  - 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.
  - 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.

Based on the user stories we developed the following flow chart (see Figure 3) whose main focus lies on the internal way of working of the online learning enabler.

The Online learning enabler works autonomically without any user interaction (cf. user stories). Before the enabler is used it can be adjusted according to the application domain. However, these adjustments are made in the backend and not through a graphical user interface. Apart from this the Online learning

enabler has some interfaces through which data is exchanged with other enablers/tools (green distorted boxes).

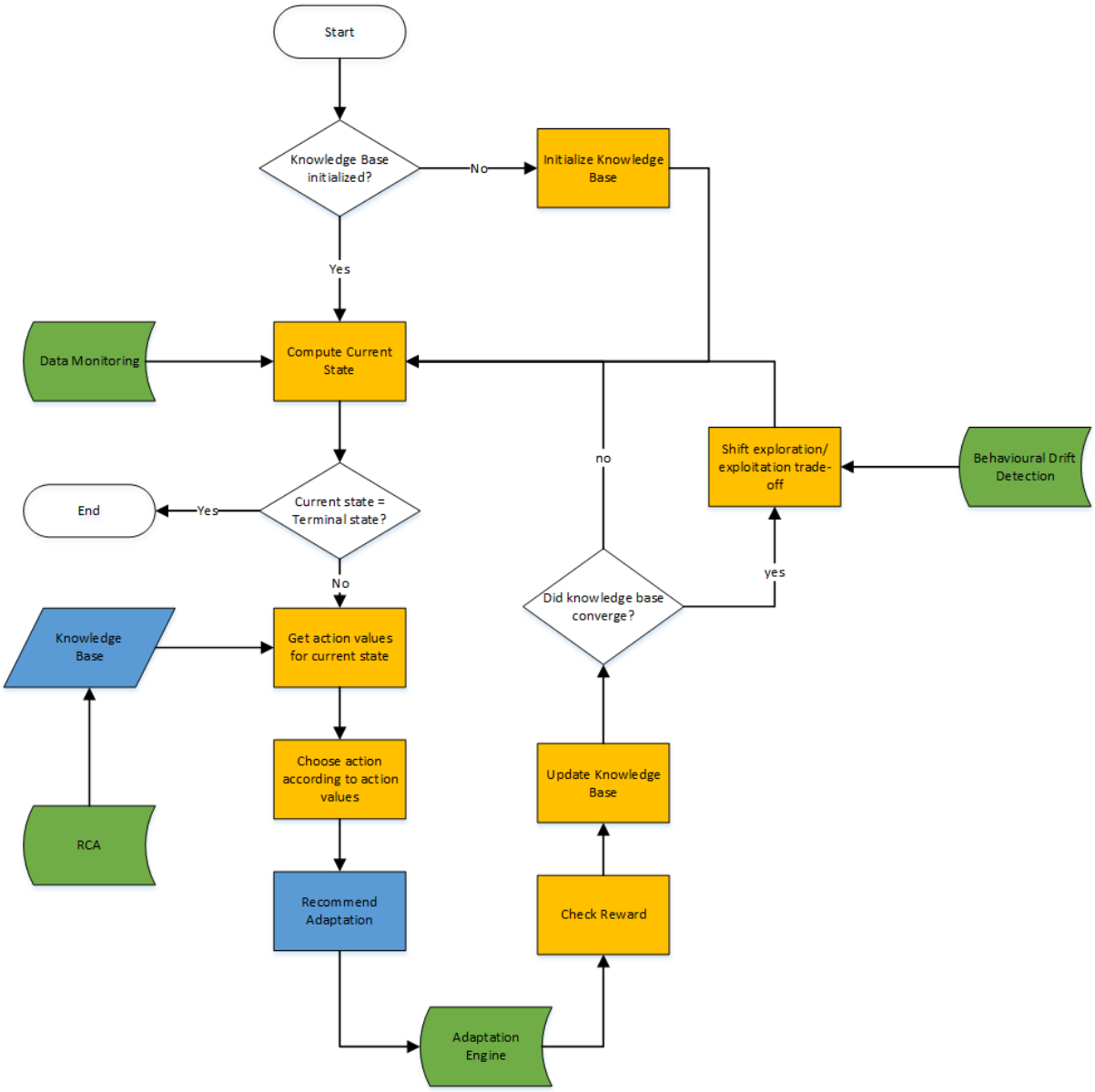

Figure 3. Online Learning flow-chart

3.1.3 Secure Orchestration and Deployment

GeneSIS (a.k.a., the Secure Orchestration and Deployment tool) aims to facilitate the engineering and continuous deployment of Smart IoT Systems (SIS), allowing decentralized processing across the IoT, edge and cloud space. GeneSIS will be agnostic to any development paradigm and technology, meaning that the developers can design and implement the applications based on their preferred paradigms and technologies. In particular, it will include:

1. **A domain specific modelling language (DSML) to model the orchestration and deployment of smart IoT systems across the IoT, edge and cloud spaces.** In addition, this language will support the specification (i) of the security mechanisms to be deployed and (ii) of metadata (e.g., software version) for each of the elements in a model.

2. **An engine to enact the orchestration and deployment of SIS on IoT, edge and cloud resources.** From a deployment model (specified using our DSML), this engine will be responsible for enacting the deployment (or adaptation) of a SIS. This will include the deployment and configuration of software components and the adaptation of a deployment (e.g., moving one software node from one host to another, installation of a new version of a software node) as well as the provisioning of cloud resources (IaaS and PaaS) and the integration with IoT middleware (e.g., SMOOL, SOFIA2).

3. **Monitor the status of a deployment.** The orchestration and deployment engine of the tool will allow tracking the status of a deployment or adaptation as well as the status of the IoT, edge, and Cloud resources once a SIS is deployed.

The user stories introduced in Listing 3 detail the features that should be offered by GeneSIS to DevOps teams.

Listing 3. GeneSIS user stories

Actors:

- IoT DevOps engineer

User stories:

- As an IoT DevOps engineer, I want to specify the deployment of my SIS, so that I can in turn automatically deploy it.
- 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.
- 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 automatically deploy a SIS,
    - so that I can automate the delivery of my system in a production environment.
    - so that I can reproduce the delivery of my system in a production environment.
    - so that I can automate the delivery of my system in a test environment.
    - so that I can reproduce the delivery of my system in a test environment.
- 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.
- 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 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.
- 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 specify the deployment of my SIS, so that I can define security mechanisms through the deployed system.

From these user stories we specified the main features that should be offered by the GeneSIS user interface together with the relationships between these features (see Figure 4. In short, the user can either start from an existing deployment model or specify a new one (see Figure 4. "Deployment Modelling" activity). This deployment modelling activity includes the tasks of specifying: (i) the types of software components to be deployed, (ii) the relationships between these software components, (iii) the host on which each software component should be deployed, (iv) the controllers to manage direct concurrent accesses to actuators, (v) a set of high level commands to facilitate the modelling activity (e.g., migrate or update software). Once the deployment model is ready, the user can save it for future uses and/or trigger a deployment. Once this deployment is completed, a process is started to monitor the systems. This process sends notifications for every change in the running system. These notifications are sent back to the user who can decide to integrate them into his model (i.e., he can update his deployment model with runtime information so that it reflects the current status of the system). Adaptation actions such as a software update or migration can be performed via a new modelling process. It is worth noting that this loop from runtime to deployment modelling provides us with the capability to perform continuous deployment.

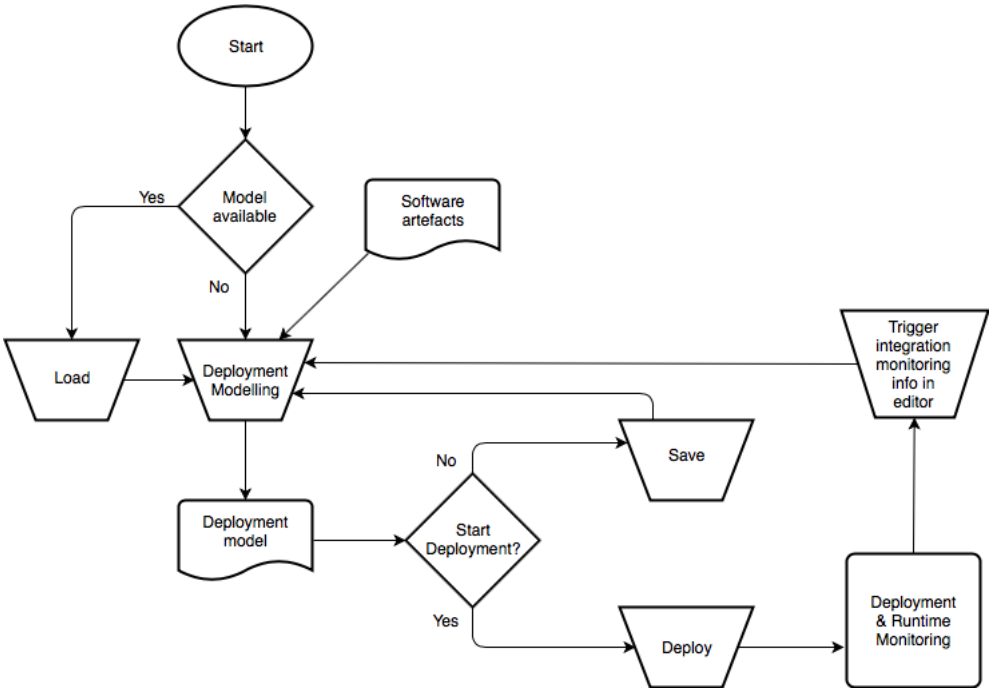

Figure 4. Flow-chart of the GeneSIS user interface

3.1.4 Actuation Conflict Manager

The main goal of the Actuation Conflict Manager (ACM) is to identify actuation conflicts present in an application and facilitate solving these conflicts. The ACM prompts the user for identified conflicts and presents him a palette of conflict resolution components that can be easily added to the application.

Detection is done by analysis of the application and using models representing the physical environment and how different actuators would interact to identify conflicts.

The palette is composed of off-the-shelf components implementing straightforward conflict management strategies but can be extended by the user if more advanced strategies are required. Each component available in the palette is complemented by metadata describing its properties, such metadata can be then used to allow filtering of the components in the palette to ease choosing the adequate strategy.

Extension of the palette is achieved either by directly writing code programming the behaviour of the conflict resolution components, or by designing the FSM (Finite State Machine) of the behaviour. In the latter case, the model will be checked to ensure certain properties before being translated into code.

Listing 4 introduces the user stories detailing the features to be offered by the actuation conflict manager.

Listing 4: Actuation conflict management user stories

Actors:

- IoT DevOps engineer

User stories:

- 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 that application and reduce misbehaviour.
    - 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 and resolve these conflicts.
    - As an IoT DevOps engineer, I want to browse a palette of pre-made actuation conflict management policies, so that I am able to know the range of solutions I have.
    - As an IoT DevOps engineer, I want to add conflict management policies to my application assembly, so that I can manage inputs to the actuators to reduce misbehaviour.
    - 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.
    - 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.
    - 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.

        - As an IoT DevOps engineer, I want to visualize the FSM used by an actuation conflict management policy, 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 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.
        - 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.
        - 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.
    - As an IoT DevOps engineer, I want to be able to design a new actuation conflict management policy using a formal model, so that I am able to implement a mathematically correct new behaviour not present in the off-the-shelf palette.
        - 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.
        - 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.
        - 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 policy available in the palette to use it in my assembly, so that I can use that new conflict management component in an application.

From these user stories we designed the flow-chart depicted in Figure 5 that shows how the ACM will work. Starting at a visualization of the application flow, conflicts will be detected and marked in the graphical user interface. Upon the user selecting a conflict, the actuation conflict management policy palette will open presenting the user with a pre-filtered list of available policies. From that window, the user can specify a different filtering query, enter the palette extension flow, or select a policy.

When a strategy is selected, the user can then visualize the actual code template implementing that strategy, visualize a formal model (if available) or choose this strategy to solve the conflict. Choosing to solve the conflict will transition to a parametrization stage where the DevOps engineer is able to tailor the policy to the specific conflict to resolve, by setting parameters like inputs or outputs. Upon validating the parametrization, the actual components are generated from the template and added to the application. As for the palette extension flow, the user will first be prompted whether he wants to write the code template for the policy, or a design the model implementing the policy. In the case of a writing template code, a simple code view with documentation for framework elements will be displayed. In the case of a policy template generated from a model, an FSM designer and validation options will be displayed. Once the code is available, either after being written by the DevOps engineer or generated from the FSM, a screen allowing the user to specify metadata will be shown before inserting the new strategy component into the palette.

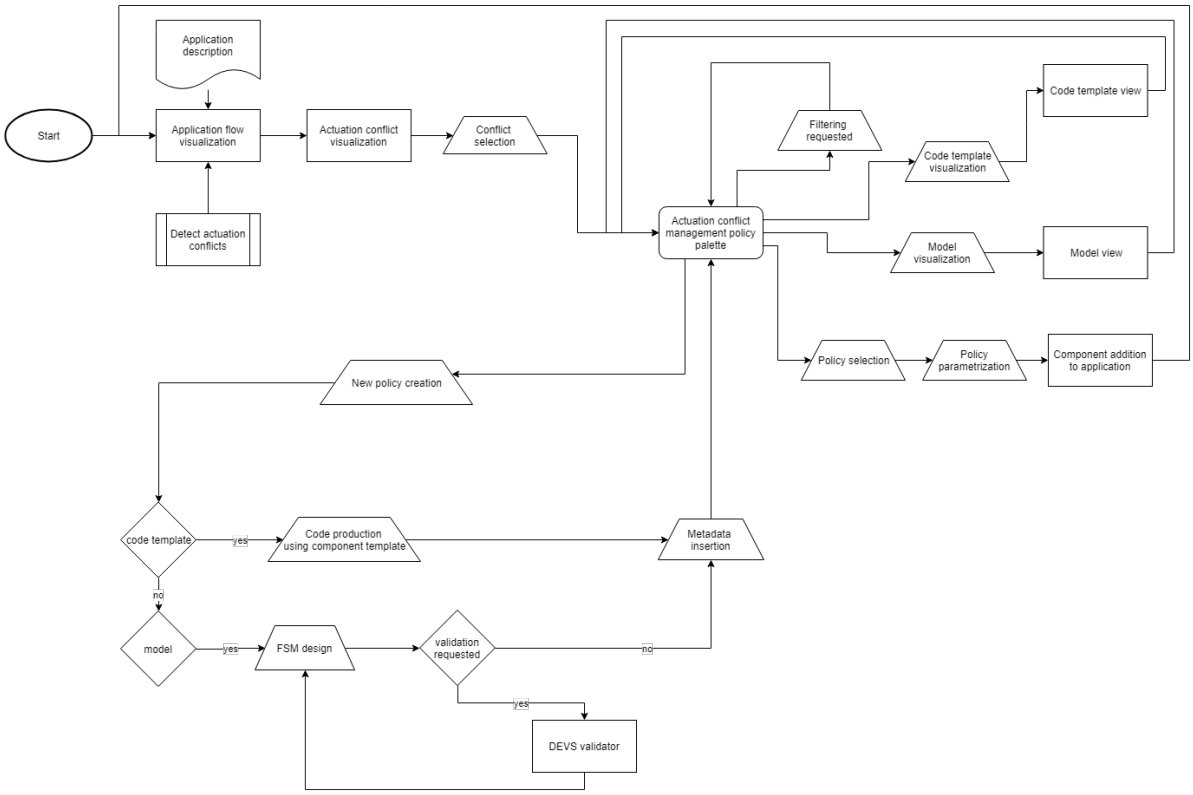

Figure 5: Actuation conflict manager flow-chart

3.1.5 Diversifier

The main objective of the Diversifier is to automatically generate diverse but functionally equivalent Smart IoT Systems, in order to improve the resilience of the system against both internal perturbations, such as the default of some devices or nodes, and external perturbations, such as security attacks.

The Diversifier provides two major features to achieve the objective: (i) it generates different implementations of the communication protocols, specified in ThingML. (ii) it synthesises diverse configurations (orchestrations) of the Smart IoT System, as specified in the GeneSIS modelling language (specifying orchestration and deployment models), by using alternative components and changing the topology of these components.

These two diversification features improve the global resilience of the IoT systems in two different ways, depending on the use cases. The users can choose to either directly deploy the diversified components or orchestrations as the actual system or use the generated configurations to increase the testing coverage of diverse configurations that may emerge during runtime.

The user stories introduced in Listing 5 detail the features that should be offered by the Diversifier to DevOps teams.

Listing 5: Diversifier user stories

Actors:

- IoT DevOps engineer

User stories:

- As an IoT DevOps engineer, I want to have multiple versions of a gateway (i.e. software to be deployed on an edge device), so that I can test all of them to assure the overall QoS.
    - I want to have multiple gateway deployment files automatically generated, so that I don't need to manually define each other.
    - I want to specify the variation points in a deployment file, so that I can decide which part will be diversified.
    - I want to specify the constraints of gateway deployments, so that the generated deployments are valid.
    - I want to have the gateways automatically deployed according to the deployment file, so that I do not need to manually deploy them.
    - I want to see how the generated gateways are different from each other, so that I can select the ones with better coverage
    - I want to run the test cases on all the generated gateways driven by the pipeline, so that I can save time on testing them manually one by one.
- As an IoT DevOps engineer, I want my IoT components to communicate in different message formats, so that they are not easily hacked.
    - I want to specify the communication protocol in an abstract, language-independent model, so that I can define the behaviour of the generated components.
    - I want to generate multiple implementations from the same specification, so that they are hard to be hacked because
        - They have different orders of parameters.
        - They have redundant parameters.
    - I want to see how the generated implementations are different from each other, so that I can select the ones with a better coverage
    - I want all the generated implementations to be automatically deployed into a valid gateway, so that I can test or operate them.
- As an IoT DevOps engineer, I want my gateway to be able to automatically change into a different deployment and configuration at runtime, so that it can recover from attacks or failures.
    - I want to change among the generated and tested deployments, so that I am confident about the quality.

The diversifier being an autonomous reasoning engine, it takes the form of a service and does not offer any graphical user interface. As depicted in Figure 6, the diversifier exposes interfaces to load variability models and the deployment models to be diversified (i.e., multiple variants of the deployment models will be generated based on the variation points specified in the variability models).

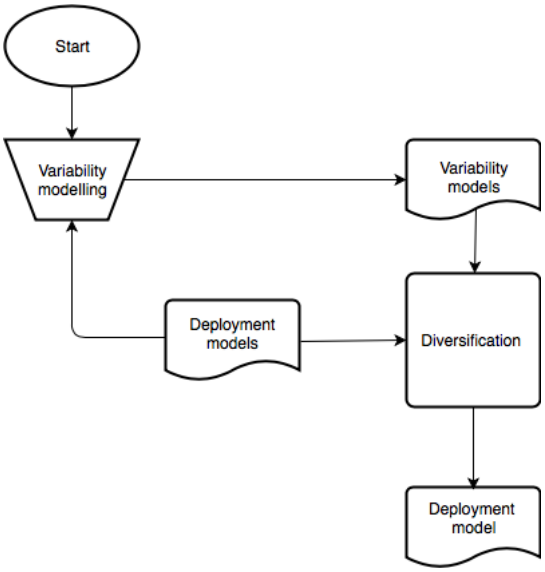

Figure 6. Flow-chart of the Diversifier user interface

3.1.6 Context-aware Access Control

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

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

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

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

Listing 6 introduces the user stories detailing the features to be offered by the Context-aware Access Control.

Listing 6: Context-aware Access Control user stories

Actors

- Device owner (ex: Patient in the Digital Health use case)
- Device user (ex: Clinician in the Digital Health use case)
- Administrator

Users 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.
- 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 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 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 be sure that all authorized people can consult the data gathered by the device to allow them performing the tasks they are responsible for.
- 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.
- As an Administrator, I want to define the level of access (i.e. different sets of data) for each person authorized to consult the data gathered by the device, to be able to define how the privacy data can be handled depending on the role of each person in the organization.
- 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.

As the Context-aware Access Control tool provides mechanisms for controlling the security, privacy and trustworthiness behaviour of Smart IoT Systems, its interaction with users is quite simple. As depicted in Figure 6, an URL is provided to the device owner to allow him to authenticate on the Context-aware Access Control tool interface, to accept or decline requested scopes for the objects by specifying a given authorization code, then to give consents to the object to use accepted scopes. These scopes and claims will be used to restrict the object accesses to backend server APIs, through a corresponding access token provided to the connected device. Then the device user accesses the data produced by the connected object in a secure way.

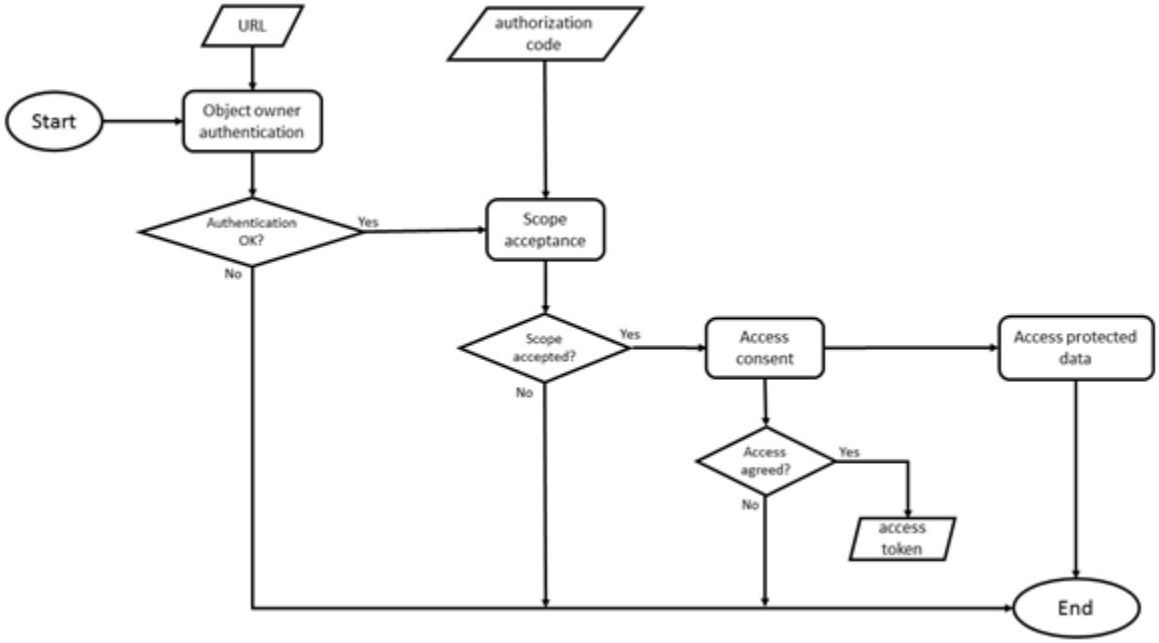

Figure 7. Flow-chart of the Context-aware Access Control tool user interface

3.1.7 Security and privacy monitoring

Once the IoT application is deployed and running, ENACT will support the operators in the DevOps team to monitor and control secure and privacy-preserving behaviour of the Smart IoT System (SIS). The security and privacy Monitoring will be in charge of capturing security behaviour information from the SIS and detecting any potential attack and security or privacy incident. The Control part is in charge of helping the operators to promptly react to detected incidents. To this end, the tool will enable them to previously configure the desired alerts and reaction models, be they automatic or manual, depending on the case.

Listing 7 introduces the user stories detailing the monitoring features to be offered by the Security and privacy monitoring and control tool.

Actors

- System operator

User stories

- As a system operator, I want to ensure the security of the IoT application at runtime, so I can verify the correct (secure) behaviour of the application.
    - As a system operator, I want to know which security capabilities of the IoT application need to be monitored, so that I can validate the secure behaviour of the application.
    - As a system operator, I want to know which security metrics apply to the capabilities of the IoT application that need to be monitored, so that I can verify the secure behaviour of the application.
    - As a system operator, I want to review the status of the security metrics of the IoT application, so that I can verify the secure behaviour of the application.

  - As a system operator, I want to establish the range of metric values corresponding with the secure behaviour of the IoT application, so I can validate the secure behaviour of the application.
  - As a system operator, I want to be notified when one of the security metrics of the IoT application has a value out of the correct (secure) range, so that I can know the application is not secure anymore.
- As a system operator, I want to monitor the network traffic of the IoT application, so I can apply security rules over the application to control the correct behaviour of the application.
  - As a system operator, I want to monitor TCP/IP protocol network traffic related to the IoT application, so I can apply security rules over the application to control the correct behaviour of the application.
  - As a system operator, I want to monitor 6LowPan protocol network traffic related to the IoT application, so I can apply security rules over the application to control the correct behaviour of the application.
  - As a system operator, I want to monitor Wi-Fi protocol network traffic related to the IoT application, so I can apply security rules over the application to control the correct behaviour of the application.
  - As a system operator, I want to monitor MQTT protocol network traffic related to the IoT application, so I can apply security rules over the application to control the correct behaviour of the application.
  - As a system operator, I want to monitor Z-Wave protocol network traffic related to the IoT application, so I can apply security rules over the application to control the correct behaviour of the application.
  - As a system operator, I want to monitor Modbus TCP protocol network traffic related to the IoT application, so I can apply security rules over the application to control the correct behaviour of the application.
  - As a system operator, I want to monitor IoT application behaviour, so I can apply security rules over the behaviour attributes to control the correct behaviour of the application
  - As a system operator, I want to check whether the communication is encrypted, so that I can know if the 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 communication is secure.
- As a system operator, I want to monitor the syslogs generated by the devices managed by the IoT application, so I can apply security rules over the application to control the correct behaviour of the application.
- As a system operator, I want to have a set of default security incidents detection rules to be selected and applied over the IoT application, so I can apply well-known incidents detection rules over the application.

Listing 7: Security and Privacy monitoring user stories

In the following flow-chart we show how monitoring, notification and reaction management will be related to one another within the tool user interface.

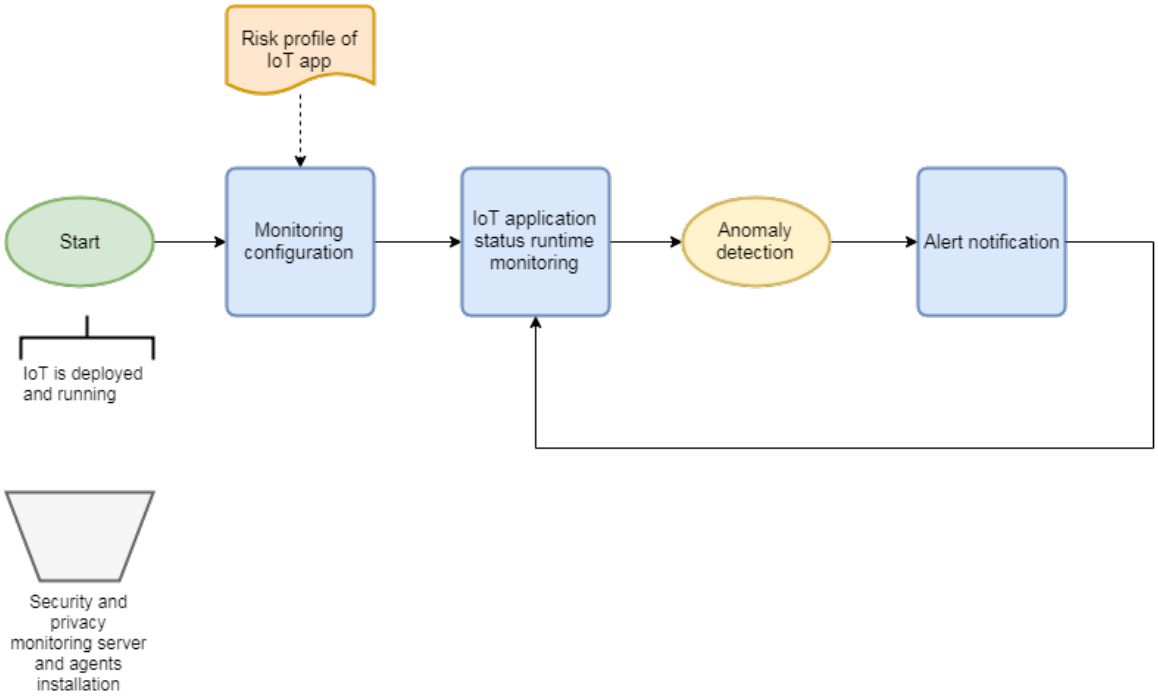

Figure 8. Flow-chart of the Security and Privacy monitoring tool user interface

3.1.8 Security and privacy control

The security and privacy control (enforcement) features offered by the tool will be managed as recommendations or automatic measures (e.g. cutting communications or enforcing the use of an Access Control mechanism) according to pre-configured reactions, as presented in the following listing.

Actors

- System operator

User stories

- As a system operator, I want to apply at runtime security controls in the IoT application.
  - As a system operator, I want to be able to cut communication in case the thing is not trustworthy.
  - As a system operator, I want to include confidentiality (e.g. by encrypting the data) into IoT application communication in case there is not in place.
  - As a system operator, I want to know which security controls I should deploy into my IoT application in order to have more secure application.
  - As a system operator, I want to enforce the use of an Access Control mechanism.

Listing 8: Security and Privacy control user stories

In the following flow-chart we show how monitoring, notification and reaction management will be related to one another within the tool user interface.

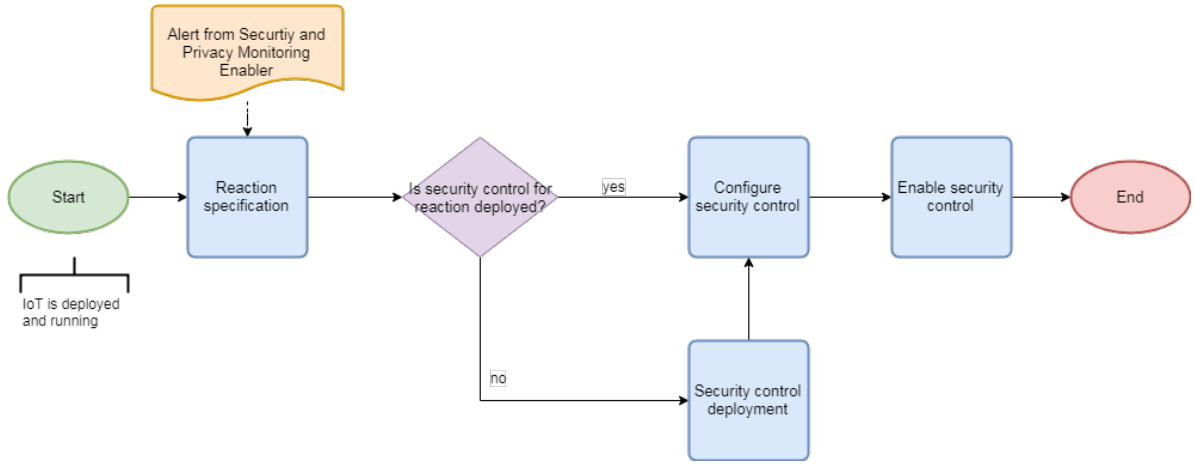

Figure 9. Flow-chart of the Security and Privacy control tool user interface

3.1.9 IoT Simulation and Emulation

The tool will be able to consume the model of the architecture of IoT devices, set of sensors deployed within the architecture and the reasoning of the actuators. The core research in the simulation of the tools is around the area of simulation of new and undetected faults based on the self-learning catalogue of faults with ability of creating new and unseen faults on the system in order to simulate system behaviour under new conditions. The tool will be able to simulate the devices and the network traffic of the devices at any scale given as an input, replicating the system structure in a production environment.

The users stories introduced in Listing 9 detail the features that should be offered by the Testing and simulation enabler.

Listing 9: Testing and simulation enabler user stories

Actors

- IoT Application Developer

User stories

- As an IoT application developer, I want to simulate my application, so that I can test the application in different scenarios.
- As an IoT application developer, I want to simulate different set of devices, so that I can ensure code cross compatibility.
- As an IoT application developer, I want to simulate huge set of devices, so that I can test how my application performs at scale.
- As an IoT application developer, I want to simulate multiple types of sensors, so that I can test foreseen data scenarios against the application to ensure application integrity.
    - As an IoT application developer, I want to simulate temperature sensors, so that I can test foreseen data scenarios against the application to ensure application integrity.
    - As an IoT application developer, I want to simulate GPS data, so that I can test foreseen data scenarios against the application to ensure application integrity.

  - As an IoT application developer, I want to simulate humidity sensors, so that I can test foreseen data scenarios against the application to ensure application integrity.
  - As an IoT application developer, I want to simulate distance sensors, so that I can test foreseen data scenarios against the application to ensure application integrity.
  - As an IoT application developer, I want to simulate pressure sensors, so that I can test foreseen data scenarios against the application to ensure application integrity.
  - As an IoT application developer, I want to simulate Accelerometer sensors, so that I can test foreseen data scenarios against the application to ensure application integrity.
  - As an IoT application developer, I want to simulate RFID data, so that I can test foreseen data scenarios against the application to ensure application integrity.
  - As an IoT application developer, I want to simulate RSSI sensors, so that I can test foreseen data scenarios against the application to ensure application integrity.
- As an IoT application developer, I want to simulate multiple types of protocols, so that I can test a complete set of IoT communication protocols.
  - As an IoT application developer, I want to simulate TCP protocol, so that I can test a complete set of IoT communication protocols.
  - As an IoT application developer, I want to simulate IR protocol, so that I can test a complete set of IoT communication protocols.
  - As an IoT application developer, I want to simulate LBT protocol, so that I can test a complete set of IoT communication protocols.
  - As an IoT application developer, I want to simulate UART protocol, so that I can test a complete set of IoT communication protocols.
  - As an IoT application developer, I want to simulate 802.15.4 (ZigBee) protocol, so that I can test a complete set of IoT communication protocols.
  - As an IoT application developer, I want to simulate MQTT protocol, so that I can test a complete set of IoT communication protocols.
- As an IoT application developer, I want to simulate cyber-attacks, so that I can ensure my application resilience, security, and privacy.
- As an IoT application developer, I want to simulate failures, so that I can ensure my application resilience, security, and privacy.
  - As an IoT application developer, I want to simulate seen scenario failures, so that I can ensure my application resilience, security, and privacy.
  - As an IoT application developer, I want to simulate unseen scenario failures, so that I can ensure my application resilience, security, and privacy.
- As an IoT application developer, I want to have control over composure of the simulation, so that I can merge simulated and real environments.

Figure 6 shows the flow-chart of the IoT Simulation covering both aspects of the enabler, the automatic actions and the manual user actions. Only trapeze actions are required from the user. The flow shows several stages of the entire process, i.e. if it starts from scratch where no device capturing data is available.

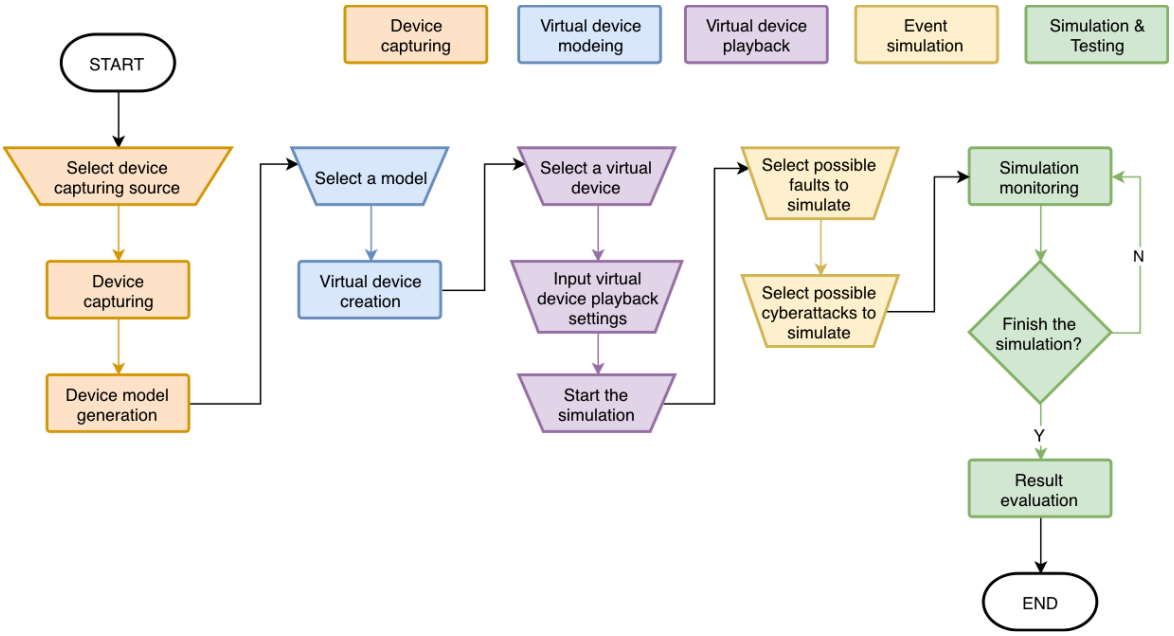

Figure 7: Flow chart of test and simulation enabler

3.1.10 Root cause analysis

The purpose of the Root cause analysis tool is to assist the System Operators while running the system. Its main capacity is to suggest potential culprits when evidences that something is not working as expected are detected, and update its suggestions based on new evidences available. Based on historical failure patterns and currently seen evidences, another important feature would be to be able to predict the evolution of an incipient problem before it actually escalates. Finally, since in large systems there are statistically many anomalies happening at the same time, a sensible prioritization between detected problems will be provided.

Listing 10: Root Cause Analysis User stories

Actors

- System Operator

User Stories

- As a System Operator, I want to know what the root-cause of an anomaly in my system is so that I can fix it faster.
- As a System Operator, I want to know the probable evolution of an anomaly in my system so that I can foresee its future impact.
- As a System Operator, if several root causes are concurrently present in the system, I want to see them prioritized by potential impact, so that I can focus on what is more relevant at each time.

Figure 8 shows the flow-chart for the Root cause analysis tool. Since it is a tool that fundamentally provides information to the user, the flow-chart depicts the internal processing of the data in order to compute the information that has to be presented to the System Operator (in the RCA dashboard).

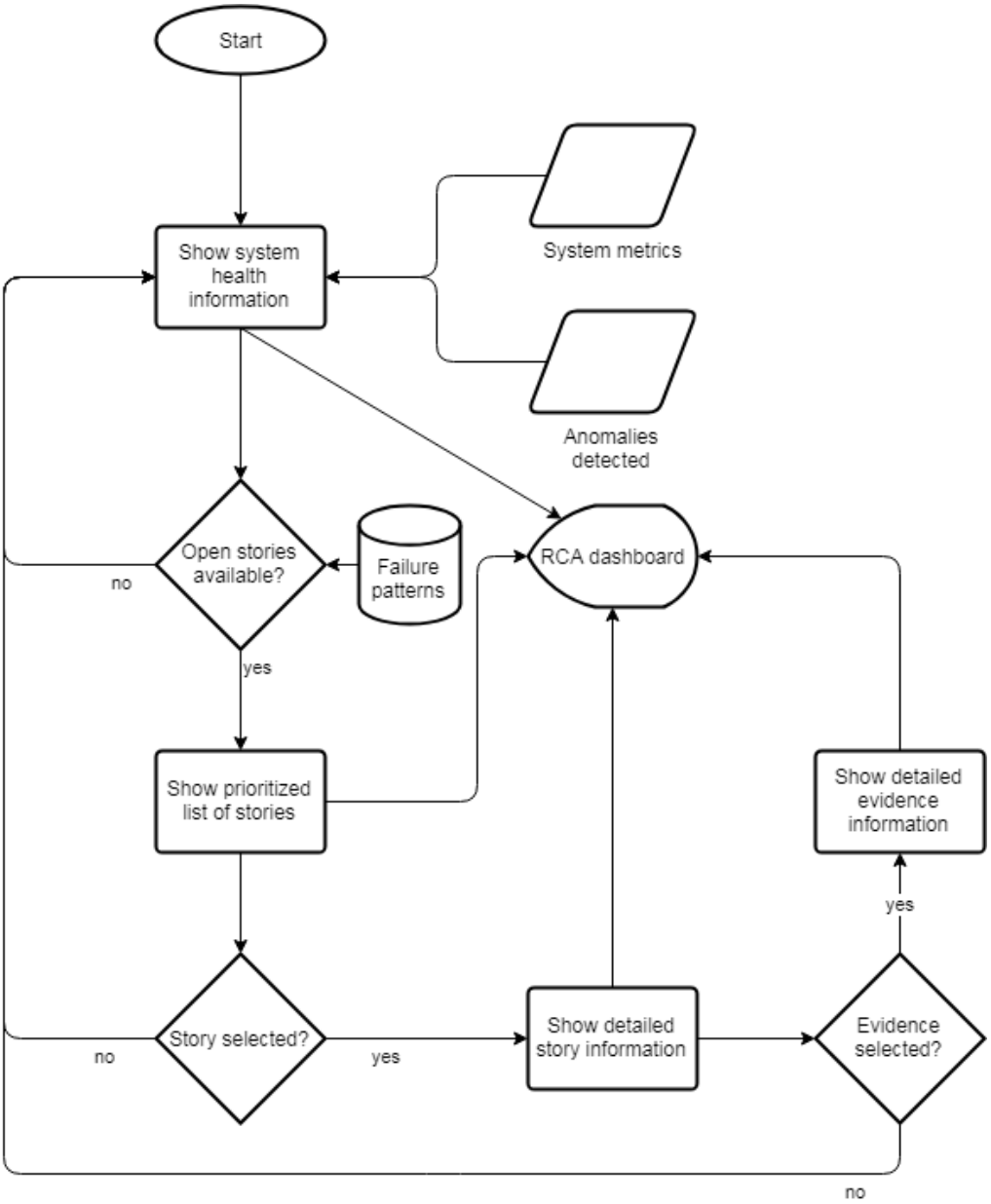

Figure 8 - Root cause analysis flow-chart

Initially the system gathers metrics and anomalies directly from the monitored elements or from other enablers, and this information is summarized in the RCA dashboard. If there are "open stories" (i.e., active problems), these open stories are prioritized and shown in the dashboard as well. If the user selects one of these stories, then extended information on it is shown, typically the actors and evidences associated to the failure. The user can select an evidence to see additional information on it (the particular type of output might depend on the type of evidence selected). All this process is iteratively executed as long as the monitoring system is active.

3.1.11 Behavioural Drift Analysis

The purpose of the Behavioural Drift Analysis is to detect whenever the application derives from its expected behaviour and monitor such drifts. Measuring the drift is done by adding an observation engine to the application, the observation engine is configured before being added, its model is either probabilistic or possibilistic and can be designed from within the deployment interface.

On the monitoring side, the system operator has access to a dashboard aggregating drift measurements from several observation engines as well as a feedback section showing him the most likely model derived from the stochastic model of the observed application.

The user stories proposed in Listing 11 define the features that the Behavioural Drift Analysis should offer.

Listing 11: Behavioural Drift Analysis User Stories

Actors:

- IoT DevOps engineer
- System operator

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 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.
    - 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 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 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 transition of the previously designed HMM, so that I can parametrize the model.
    - As an IoT DevOps engineer, I want to set the values on each state of the previously designed HMM, 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.
- As an IoT DevOps engineer I want to use a graphical designer to input a possibilistic 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 HMM 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 transition of the previously designed HMM, 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.

Figure 9 details the flow-chart of the behavioural drift analysis from the DevOps engineer's point of view. From the start, the DevOps engineer selects in a list of applications the one he wishes to add a new observation engine to. After picking an application, the user is asked to choose which type of model will be used by the observation engine. Then a few screens to design and parametrize the model are shown that upon validation will export the configuration to a format understood by the observation engine core, which will then be added to the application.

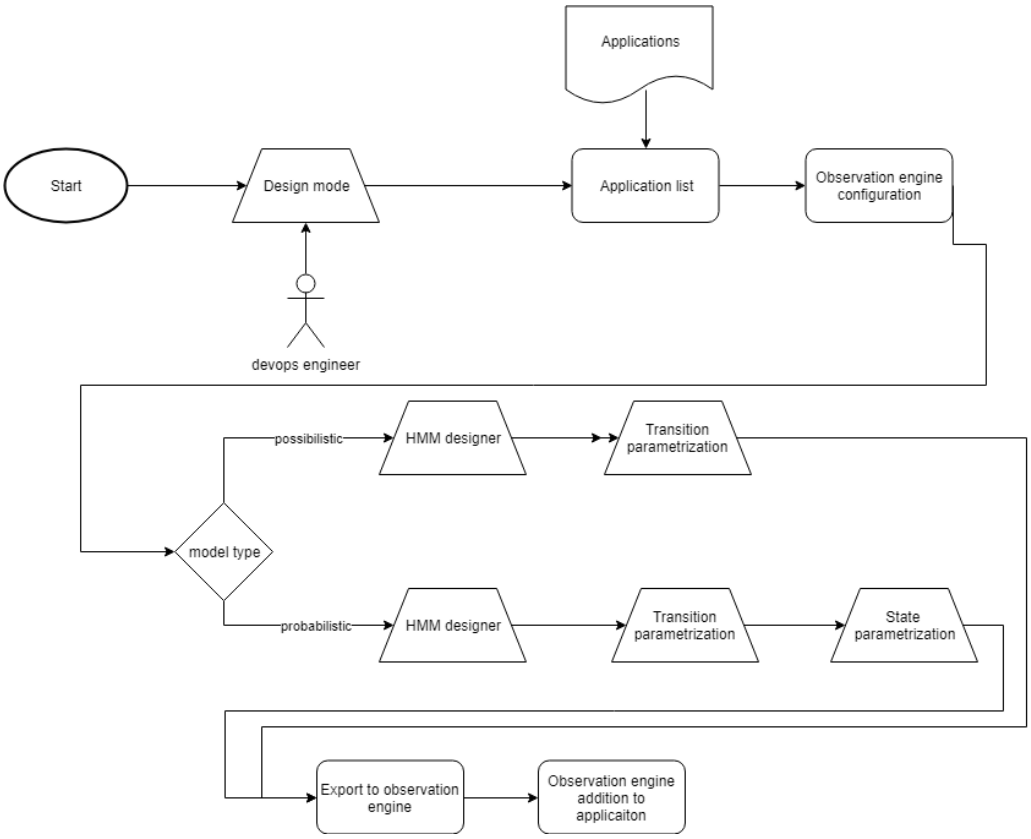

Figure 9: Behavioural drift analysis flow-chart from a DevOps engineer's point of view

Figure 10 shows the flow-chart of the behavioural drift analysis from the system operator's point of view. The system operator is presented with a dashboard aggregating measurements from the installed observation engines, as well as the most likely model.

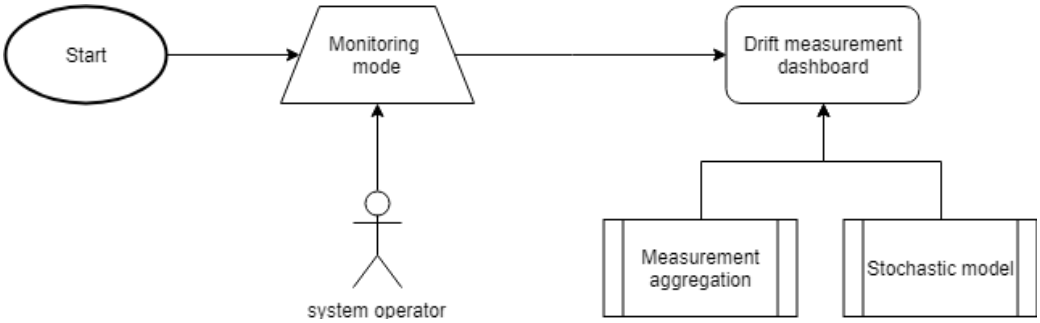

Figure 10: Behavioural drift analysis flow-chart from a system operator's point of view

4 Integration tools

Within this section we will go over the tools and the standards which has been selected to streamline and unify the ENACT project outcomes. All tools produced as a result of ENACT shall follow the proposed approach and be hosted within the defined repository.

4.1 Repository

Following the DevOps convention of the tools delivery, ENACT technical outcomes will be available on the publicly accessible repository for all the open-source enablers and connected software.
ENACT consortium chose Gitlab[11] as the host of the repository requirements the project had. Gitlab also supports Continuous Integration and Continuous Delivery of the repositories as well as the status board where the use case testing will be performed.

ENACT repository group has been created in the publicly accessible GitLab instance and it is available under url: https://gitlab.com/enact. As of the time of this writing, there are 13 repositories created for various enablers of the project, as showcased in Figure 11. Following the M15 initial release, the group will be marked as publicly available and the repositories will be available.

[11] https://gitlab.com/

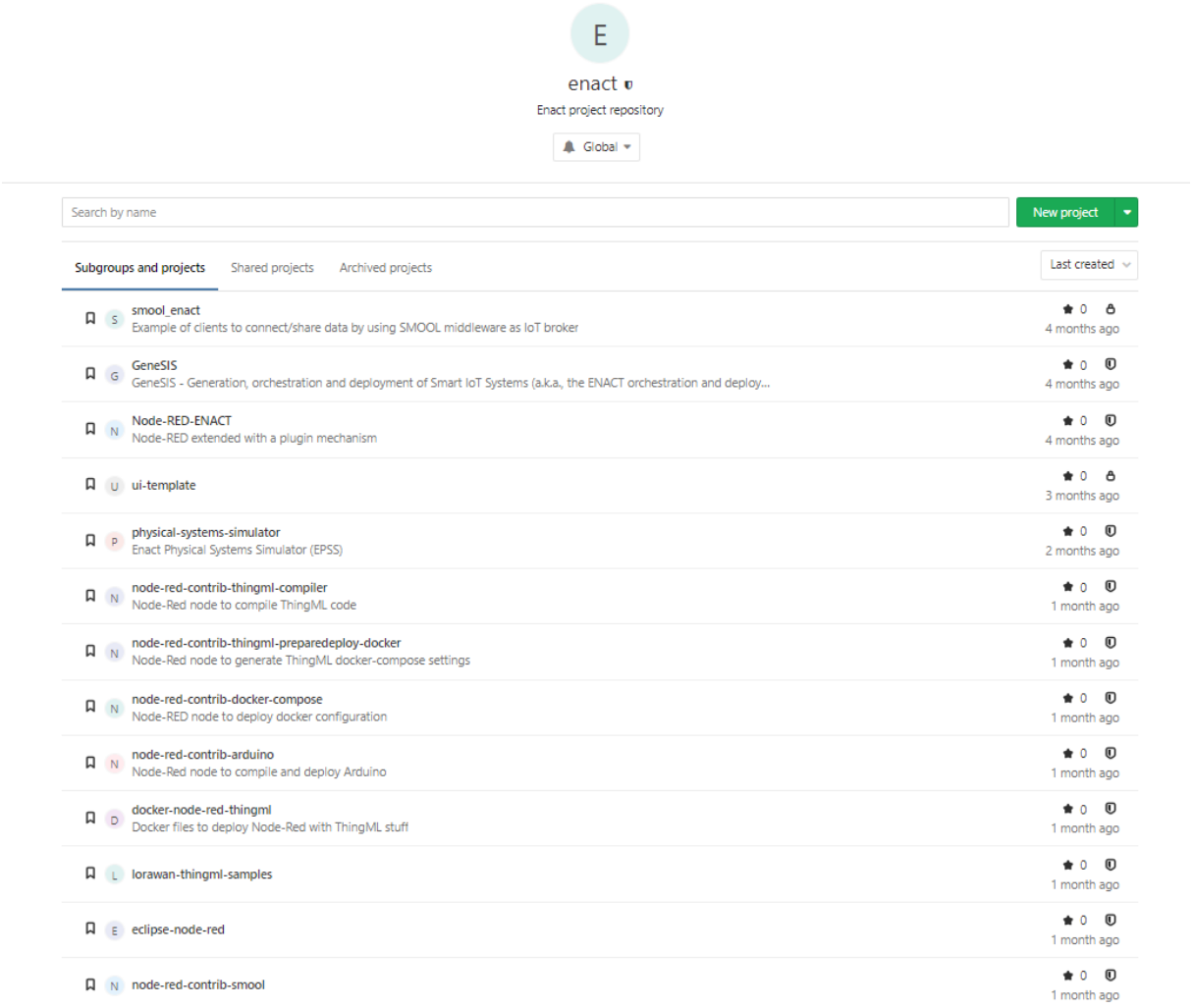

Figure 11. Enact Repository Group

4.2 Unified Enact API standards

ENACT aims to provide the easy to integrate DevOps solution which imposes a set of best practices and standards on the way the enablers will be communicating with possible integrators. Minimal set of requirements has been gathered in order to standardise the ENACT tooling, these are:

- The ENACT enablers must handle cross schema outputs (handling JSON and XML).
- The standard chosen needs to allow for the tool developers to be easily documentable.
- The ENACT enablers must be following a commonly accepted and used standard.
- The ENACT enablers must support handling of the data schema discovery.

Following these requirements two main standards have been chosen:

1. **White House API standard**[12]: widely accepted within the DevOps community, providing clear understanding of the requirements for the tool developers. Within the ENACT enablers it will be used for non-data intensive REST operations, action triggers etc.

[12] https://github.com/WhiteHouse/api-standard

2. **ODATA v4**[13]: widely known, proven within the market to be one of the best data intensive handling API standard.

In order to drive the adoption and allow the data schema discovery for the inner but also outer communication of the ENACT framework enablers, all APIs will be documented with Swagger[14], which does provide a complete suite of API discovery and testing. This will also enable ENACT to showcase the best DevOps practices as all the API documentation of the project as well as the description and testing will be carried out by usage of the Swagger toolset.

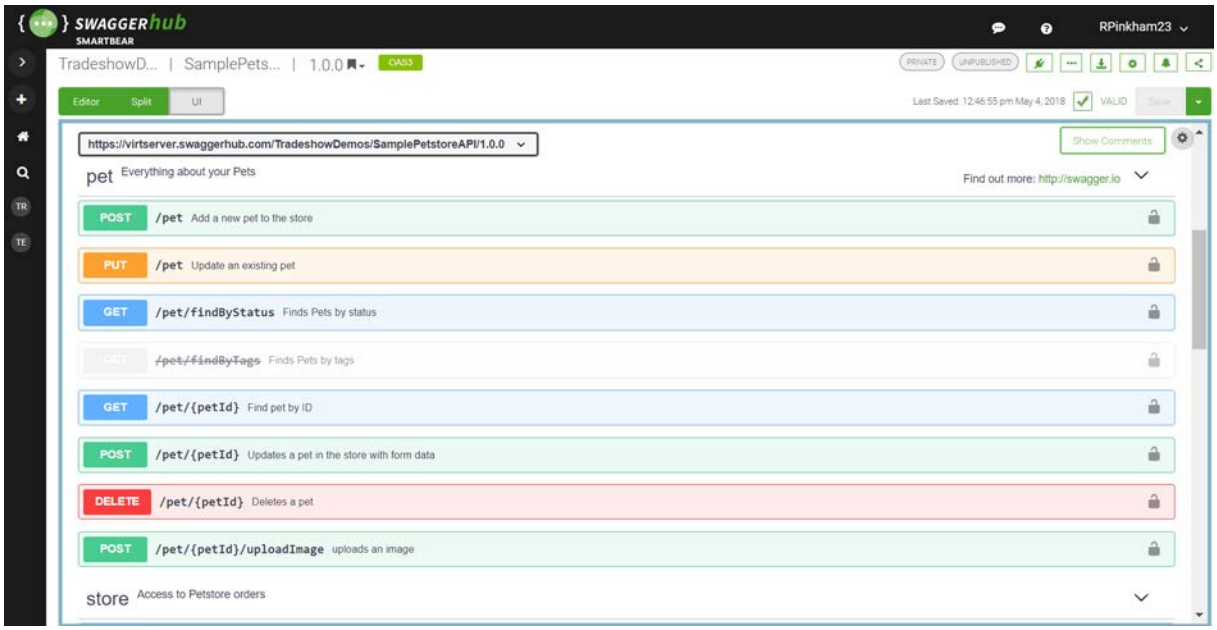

Figure 12. Swagger UI

4.3 Unified Enact UI

ENACT aims to provide a set of tools which does require a level of integration beyond the communication channel. A set of requirements were established in order to choose the best possible technology which would help the projects outcomes among to be understood and well received within the DevOps community. The requirements of the UI framework used within ENACT are:

- The technology the UI framework uses needs to be up to date.
- It needs to be backed up by a big vendor as opposite to high variety high velocity open source projects.
- Web components enabled.
- Flexbox enabled. Flexbox is a new web layout paradigm which ensures that the content is always readable and removes the cost of targeting various device screen sizes.

Based on these requirements ENACT consortium will use an open-source ReactJS web framework called Mineral-UI[15]. This framework is developed by CA Technologies and built upon the very popular web framework developed by Facebook. ENACT consortium has guaranteed access to the UI framework as well as guaranteed access to influence the offering of the framework. CA Technologies is

[13] https://www.odata.org/documentation/
[14] https://swagger.io/
[15] https://mineral-ui.com/

committed to provide all necessary support and additional UI elements development in order to accommodate the ENACT framework needs.

On top of these actions, a baseline template has been developed which wraps all the technologies necessary for the enablers developing parties to jump start the development of the tools in a coherent and state of the art fashion. Current version of the UI template can be found in the project GitLab page under url: https://gitlab.com/enact/ui-template . From the M18 onwards, this page will be publicly accessible without the need for the log in. Until this time, the repositories are configured in per invite access only.

4.4 Planning processes and delivery

In order to gain competitive advantage in terms of the project exploitation, ENACT consortium agreed to follow a lean approach to the project delivery. This means no less that the initial version of the tools should already be able to showcase the MVV status. This is scheduled on the close of the M15 of the project.

On top of that, further developments of the tools shall enhance the experience, allowing for even shorter MVV and better integration of the tools without introducing breaking changes into the system.
Leveraging methods and tools described in the Section 2.2 should enable the project to produce better results faster, which is one of the principals of Agile DevOps operations.

To conclude, this deliverable highlighted the means, methods and plan for the technical vision of the ENACT project, the methods used for the results development and delivery and the agile approach for the research the ENACT consortium is currently using to guarantee the best results possible.