



Rational decomposition and orchestration for serverless computing

Deliverable D2.1

Initial requirements and baselines

- Companion document

Version: 1.0

Publication Date: 29-June-2019

Disclaimer:

The RADON project is co-funded by the European Commission under the Horizon 2020 Framework Programme. This document reflects only authors' views. EC is not liable for any use that may be done of the information contained therein.

Deliverable Card

Deliverable	D2.1
Title:	Initial requirements and baselines
Editor(s):	Satish Srirama (UTR)
Contributor(s):	Matej Artač (XLB), Matija Cankar (XLB), Giuliano Casale (IMP), Stefania D'Agostini (ENG), Chinmaya Dehury (UTR), Martin Garriga (TJD), Pelle Jakovits (UTR), Hans Georg Næsheim (PRQ), Dušan Okanović (UST), Satish Srirama (UTR), Damian Tamburri (TJD), Vasilis Tountopoulos (ATC), Andre van Hoorn (UST), Michael Wurster (UST), Vladimir Yussupov (UST), Lulai Zhu (IMP).
Reviewers:	Matija Cankar (XLB), Vasilis Tountopoulos (ATC)
Type:	R
Version:	1.0
Date:	29-June-2019
Status:	Final
Dissemination level:	Public
Download page:	http://radon-h2020.eu/public-deliverables/
Copyright:	RADON consortium

The RADON project partners

IMP	IMPERIAL COLLEGE OF SCIENCE TECHNOLOGY AND MEDICINE
TJD	STICHTING KATHOLIEKE UNIVERSITEIT BRABANT
UTR	TARTU ULIKOOL
XLB	XLAB RAZVOJ PROGRAMSKE OPREME IN SVETOVANJE DOO
ATC	ATHENS TECHNOLOGY CENTER ANONYMI BIOMICHANIKI EMPORIKI KAI TECHNIKI ETAIREIA EFARMOGON YPSILIS TECHNOLOGIAS
ENG	ENGINEERING - INGEGNERIA INFORMATICA SPA
UST	UNIVERSITAET STUTTGART
PRQ	PRAQMA A/S

The RADON project (January 2019 - June 2021) has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 825040

Executive summary

This is the companion document to the RADON Initial requirements and baselines deliverable (**D2.1**). The document lists the requirements for the RADON tools as well as for the three industrial use case scenarios. Detailed information of the RADON requirements is provided with respect to different work packages.

Glossary

CDL	Constraint Definition Language
FaaS	Function-as-a-Service
GMT	Graphical Modelling Tool
NLP	Natural Language Processing
TOSCA	Topology and Orchestration Specification for Cloud Applications

Table of contents

1.	Introduction	6
1.1.	Structure of the document	6
2.	WP2: Integrated Framework & Requirements	6
3.	WP3: Methodology and Quality Assurance Requirements	13
4.	WP4: Modelling Environment Requirements	29
5.	WP5: Runtime Environment Requirements	40
6.	Use Case 1: Ambient Assisted Living UC	49
7.	Use Case 2: Managed DevOps UC	52
8.	Use Case 3: Travel Technology UC	56

1. Introduction

This RADON companion document contains the detailed information regarding the RADON requirements with respect to different work packages. This document lists the requirements for the RADON tools as well as for the three industrial use case (UC) scenarios: Ambient Assisted Living UC, Managed DevOps UC, and Travel Technology UC.

1.1. Structure of the document

The document is structured as follows.

- Section 2 provides the requirements of the RADON Integrated Framework.
- Section 3 provides the Methodology and Quality assurance requirements.
- Section 4 provides the requirements of the RADON Modelling Environment.
- Section 5 provides the requirements of the RADON Runtime Environment.
- Section 6 provides the requirements of the Ambient Assisted Living use case.
- Section 7 provides the requirements of the Managed DevOps use case.
- Section 8 provides the requirements of the Travel Technology use case.

2. WP2: Integrated Framework & Requirements

Note: Unused fields are not shown in the corresponding table.

ID:	R-T2.3-1
Type:	FUNCTIONAL_SUITABILITY
User story:	As a Software Developer, I want to develop application in a parallel manner
Requirement:	The IDE must be able to support development by teams that work in parallel on the application
Extended Description:	Several developers will be able to work on the same application without the need to install software.
Priority:	Must have
Affected Tools:	IDE
Means of Verification:	Collaboration between team members
Dependency:	R-T2.3-6

ID:	R-T2.3-2
Type:	FUNCTIONAL_SUITABILITY
User story:	As a Software Developer, I want to develop application using Java, Go, Node.js, Python and Ruby programming languages
Requirement:	The IDE must be able to support the use of Java, Go,

	Node.js, Python and Ruby programming languages
Extended Description:	Even if the IDE must not lock-in to particular language, the Java, Go, Node.js, Python and Ruby programming languages must be supported
Priority:	Must have
Affected Tools:	IDE
Means of Verification:	Direct implementation on IDE

ID:	R-T2.3-3
Type:	FUNCTIONAL_SUITABILITY
User story:	As a Software Developer, I want to provide new capabilities through custom plug-in development
Requirement:	The IDE must be extensible by customizing built-in plugins.
Extended Description:	The provision of new editors/wizards, new API's and workspace agents plugins are examples of customization
Priority:	Must have
Affected Tools:	IDE
Means of Verification:	Extensibility and customization of the IDE

ID:	R-T2.3-4
Type:	FUNCTIONAL_SUITABILITY
User story:	As a Software Developer, I want to step through the debugging of the application and identify errors
Requirement:	The IDE must provide debugger functionalities
Extended Description:	Debugging facilities will allow to navigate the thread of the application code through common debug operations (i.e. variable watching and substitutions, breakpoints, step into/over)
Priority:	Must have
Affected Tools:	IDE
Means of Verification:	Inspection of application code step-by-step

ID:	R-T2.3-5
Type:	FUNCTIONAL_SUITABILITY
User story:	As a Software Developer, I want to develop applications using source code editors which simplify and speed up the typing of source code
Requirement:	The IDE must provide source code editors supporting syntax highlighting, auto-complete and error checking capabilities
Extended Description:	The writing and editing of source code will be enhanced and simplified through source code editors with syntax highlighting and auto-complete functionalities
Priority:	Must have
Affected Tools:	IDE

Means of Verification:	Direct implementation on IDE
-------------------------------	------------------------------

ID:	R-T2.3-6
Type:	SECURITY
User story:	As a Software Developer, I want to secure application workspaces from unauthorized users
Requirement:	To secure application and services the IDE must provide identity and access management functionalities
Extended Description:	Authentication and authorization management capabilities will be applied to regulate access to different IDE's entities (e.g. workspaces, stacks etc..)
Priority:	Must have
Affected Tools:	IDE
Means of Verification:	Support of access control policies

ID:	R-T2.3-7
Type:	FUNCTIONAL_SUITABILITY
User story:	As a QoS Engineer, I want to be able to maintain my continuous tests.
Requirement:	The IDE must provide support in launching the TESTING_TOOL and to trigger the execution of tests
Extended Description:	The TESTING_TOOL will be integrated on the IDE in order to support the annotation of the RADON models with test-related information
Priority:	Must have
Affected Tools:	IDE, TESTING_TOOL
Means of Verification:	Tests and demo based on the case study scenarios
Dependency:	R-T3.3-6, R-T3.3-7, R-T.3.3-14, R-T.3.3-14

ID:	R-T2.3-8
Type:	FUNCTIONAL_SUITABILITY
User story:	As a QoS Engineer, I want to receive test results in the form of a comprehensive report.
Requirement:	The IDE must provide access to a report based on the results received from the TESTING_TOOL
Extended Description:	The IDE must be able to retrieve the results of the tests from the TESTING_TOOL and show them with a graphical representation and textual explanation of the data
Priority:	Must have
Affected Tools:	IDE, TESTING_TOOL
Means of Verification:	Tests and demo based on the case study scenarios
Dependency:	R-T3.3-5, R-T3.3-7, R-T.3.3-13

ID:	R-T2.3-9
------------	----------

Type:	FUNCTIONAL_SUITABILITY
User story:	As a Software Developer, I want to store and share the application code and the RADON models in a repository
Requirement:	The IDE must provide access to the shared repositories of code and RADON models
Extended Description:	The shared repositories of code and RADON models will reside externally to the IDE and will be accessed by developers through appropriate tools (e.g. GIT)
Priority:	Must have
Affected Tools:	IDE
Means of Verification:	Access to the shared repositories

ID:	R-T2.3-10
Type:	FUNCTIONAL_SUITABILITY
User story:	As a Software Developer, I want to progressively refine the application model and the application code through periodic commit on the source repositories
Requirement:	The IDE must provide connectors to CI/CD tools in order to enable the continuously release of applications via the automatic management of repository update notifications
Extended Description:	Once received a notification about the repository update the CI/CD tools will manage it by fetching the updates from the repository and performing compilation and assembly operation of source code. The IDE will provide connectors to CI/CD pipelines to automate application releases.
Priority:	Must have
Affected Tools:	IDE, DELIVERY_TOOLCHAIN
Means of Verification:	Tests and demo based

ID:	R-T2.3-11
Type:	FUNCTIONAL_SUITABILITY
User story:	As a Software Developer, I want to model deployment configuration to automatically generate deployment scripts
Requirement:	The IDE must support the deployment process in order to automatically generate and execute deployment scripts
Extended Description:	The IDE will support the deployment of FaaS-based applications on the target clouds
Priority:	Must have
Affected Tools:	IDE, DELIVERY_TOOLCHAIN
Means of Verification:	Tests and demo based

ID:	R-T2.3-12
Type:	FUNCTIONAL_SUITABILITY
User story:	As a Software Developer, I want to create a RADON-based workspace

Requirement:	The IDE must provide the opportunity to create a new RADON-based workspace
Extended Description:	The RADON-based workspace will provide a "stack" having all the tools, plugins and extensions enabled (e.g. Theia IDE, cloning of TOSCA type repo, Winery, Verification Tool, Decomposition Tool, etc.)
Priority:	Must have
Affected Tools:	IDE
Means of Verification:	Availability of a RADON-based workspace in the IDE

ID:	R-T2.3-13
Type:	FUNCTIONAL_SUITABILITY
User story:	As a Software Developer, I want to create a RADON modelling project
Requirement:	The IDE must provide the opportunity to create a new RADON modelling project
Extended Description:	The RADON modelling project will provide a directory structure complaint with the graphical modeling tool (GMT, Winery).
Priority:	Must have
Affected Tools:	IDE
Means of Verification:	Availability of a RADON modelling project type in the IDE

ID:	R-T2.3-14
Type:	FUNCTIONAL_SUITABILITY
User story:	As a Software Developer, I want to launch the GMT from a RADON modelling project
Requirement:	The IDE must provide support to launch the GMT from a RADON modelling project
Extended Description:	The GUI provided by the graphical modeling tool (GMT, Winery) will be integrated into the IDE so that will be possible to launch the GMT based on a specific workspace containing a RADON modelling project.
Priority:	Must have
Affected Tools:	IDE, GMT
Means of Verification:	Integration of the GMT into the IDE and demo based
Dependency:	R-T4.3-1
Conflict:	N/A

ID:	R-T2.3-15
Type:	FUNCTIONAL_SUITABILITY
User story:	As a Software Developer, I want to access projects inside a workspace from the GMT
Requirement:	The IDE must support the integration of a Winery plugin in

	order to navigate from the GMT to the respective workspace where the source code is maintained
Extended Description:	Winery/GMT will need to register a plugin in order to interact with a workspace (e.g. to open a respective source code editor if someone wants to edit an attached TOSCA code artifact)
Priority:	Must have
Affected Tools:	IDE, GMT
Means of Verification:	Integration of the GMT into the IDE and demo based
Dependency:	R-T4.3-2, R-T4.3-3

ID:	R-T2.3-16
Type:	FUNCTIONAL_SUITABILITY
User story:	As a Software Developer, I want to visualize changes made within the IDE in the GMT and vice versa
Requirement:	The IDE must support the synchronization of changes made from within IDE with the GMT and vice versa
Extended Description:	Changes made from within IDE will be synchronized with GMT and vice versa, e.g. when function's source code or a TOSCA definition is changed directly in the workspace, GMT has to highlight that the corresponding entity was modified
Priority:	Must have
Affected Tools:	IDE, GMT
Means of Verification:	Integration of the GMT into the IDE and demo based

ID:	R-T2.3-17
Type:	PORTABILITY
User story:	As a Software Developer, I want to export a project for reuse it in other environments
Requirement:	The IDE must provide the possibility to export projects
Extended Description:	The IDE will allow to export the chosen projects for reuse them in other environments
Priority:	Must have
Affected Tools:	IDE
Means of Verification:	Provision of export capabilities

ID:	R-T2.3-18
Type:	FUNCTIONAL_SUITABILITY
User story:	As a Software Developer, I want to launch the TOSCA management UI from a RADON modelling project
Requirement:	The IDE must provide support in launching the TOSCA management UI from a RADON modelling project
Extended Description:	The IDE will provide support to launch the TOSCA

	management UI provided by Winery from a RADON modelling project inside the workspace in order to edit the related TOSCA entities
Priority:	Must have
Affected Tools:	IDE, GMT
Means of Verification:	Integration of the GMT into the IDE and demo based

ID:	R-T2.3-19
Type:	PORTABILITY
User story:	As a Software Developer, I want to export the application blueprints as TOSCA CSAR
Requirement:	The IDE must provide the possibility to export the application blueprints as TOSCA CSAR
Extended Description:	Using a workspace plugin will be possible to export the application blueprints as TOSCA CSAR
Priority:	Must have
Affected Tools:	IDE
Means of Verification:	Export capability of application blueprints as TOSCA CSAR

ID:	R-T2.3-20
Type:	FUNCTIONAL_SUITABILITY
User story:	As a Software Developer, I want to use a source code editor with support for TOSCA grammar
Requirement:	The IDE should provide a source code editor with support for the TOSCA grammar
Extended Description:	The IDE should provide a source code editor with syntax highlighting and code completion for the TOSCA grammar
Priority:	Should have
Affected Tools:	IDE
Means of Verification:	Availability in the IDE of a source code editor with TOSCA grammar support

ID:	R-T2.3-21
Type:	FUNCTIONAL_SUITABILITY
User story:	As a Software Developer, I want to verify the quality of the developed code (i.e. Infrastructure-as-Code templates/blueprints)
Requirement:	The IDE must provide support to launch the DEFECT_PRED_TOOL and to show the state of the infrastructure in a graphical way
Extended Description:	The DEFECT_PRED_TOOL will be integrated on the IDE in order to support the identification of IaC defects
Priority:	Must have
Affected Tools:	IDE, DEFECT_PRED_TOOL

Means of Verification:	Integration of the DEFECT_PRED_TOOL into the IDE and demo based
Dependency:	R-T3.4-1, R-T3.4-2

ID:	R-T2.3-22
Type:	FUNCTIONAL_SUITABILITY
User story:	As an Operations Engineer/Release Manager I want to monitor the status of the deployment process
Requirement:	The IDE must provide support in showing the monitoring information collected by the MONITORING_SYSTEM
Extended Description:	The IDE will connect to a dashboard to keep users updated on the status of the deployment stages
Priority:	Must have
Affected Tools:	IDE, MONITORING_SYSTEM
Means of Verification:	Connection to the monitoring information from the IDE and demo based
Dependency:	R-T5.1-3

ID:	R-T2.3-23
Type:	USABILITY
User story:	As a Software Developer/Operations Engineer/QoS Engineer I want to launch the RADON tools via a dedicated menu
Requirement:	The IDE must provide a menu in order to use the different RADON tools
Extended Description:	The IDE will provide a RADON menu from which the integrated RADON tools can be launched
Priority:	Must have
Affected Tools:	IDE
Means of Verification:	Availability of the RADON Menu and demo based

3. WP3: Methodology and Quality Assurance Requirements

ID:	R-T3.1-1
Type:	FUNCTIONAL_SUITABILITY
User story:	As a Software Developer I want the framework to support me throughout the operations of data gathering, filtering, transformation and analysis
Requirement:	The methodology must define and support the pipelines needed for data ingestion, filtering, transformation, analysis, and transfer
Extended Description:	The methodology will define and support the operations of abstracting and orchestrate the pipelines needed for data

	ingestion, filtering, transformation, analysis, and transfer (e.g., as the ones managed by platforms such as AWS Data pipeline or Apache NiFi).
Priority:	Must have
Affected Tools:	DELIVERY_TOOLCHAIN

ID:	R-T3.1-2
Type:	COMPATIBILITY
User story:	As a Software Developer I want to use the framework to remain as compatible as possible to TOSCA
Requirement:	Description and orchestration of Faas-based microservices and data pipelines must remain compatible with existing constructs in TOSCA.
Priority:	Must have
Affected Tools:	FAAS_ABS_LAYER
Dependency:	R-T3.1-1

ID:	R-T3.1-3
Type:	FUNCTIONAL_SUITABILITY
User story:	As a Software Developer I want to integrate RADON's tools to support my operations
Requirement:	Every tool must provide a clear interaction and integration sequence with respect to the rest of RADON tools (to increase reusability, modularity and composition).
Priority:	Must have
Affected Tools:	ALL_TOOLS

ID:	R-T3.1-4
Type:	FUNCTIONAL_SUITABILITY
User story:	As a Software Developer I want to customize my pipeline by means of the toolchain
Requirement:	The delivery toolchain must be customizable
Extended Description:	The delivery toolchain must be customizable in such a way to allow the user to specify which tools include and in which order execute them.
Priority:	Must have
Affected Tools:	DELIVERY_TOOLCHAIN
Dependency:	R-T3.1-3

ID:	R-T3.1-5
Type:	PERFORMANCE_EFFICIENCY

User story:	As a Software Developer I want to use RADON's pipelines in parallel to save time
Requirement:	Pipelines could execute in parallel.
Priority:	Must have
Affected Tools:	DELIVERY_TOOLCHAIN

ID:	R-T3.1-6
Type:	SECURITY
User story:	As a QoS Engineer I want the framework to give access to sensitive data and services only to the right team.
Requirement:	Security and privacy policies must be automatically instantiated in the runtime environment to ensure that only the right team have access to sensitive data and services.
Priority:	Must have
Affected Tools:	ORCHESTRATOR

ID:	R-T3.4-1
Type:	FUNCTIONAL_SUITABILITY
User story:	As an Operations Engineer I want the tool to be integrated with the IDE and other tools in RADON
Requirement:	The defect-prediction tool must provide APIs to be easily integrated with other tools in RADON and with the IDE
Extended Description:	<p>The tool must be integrated with other tools in RADON as well as plugin in popular IDEs such as Eclipse (e.g., Eclipse Che).</p> <p>Mainly, the requirement will be fulfilled to make sure the defect-prediction tool to be integrated with Winery/IDE as well as the delivery toolchain</p>
Priority:	Must have
Affected Tools:	DEFECT_PRED_TOOL
Means of Verification:	Direct implementation on IDE, feature checklist, case-study

ID:	R-T3.4-2
Type:	USABILITY
User story:	As an Operations Engineer I want to use the tool with a Graphical User Interface
Requirement:	The defect-prediction tool must display a GUI for the plugin (to be integrated in the IDE)
Extended Description:	The tool should provide information about the state of the infrastructure (e.g. listing vulnerable nodes) in a graphical fashion.
Priority:	Must have

Affected Tools:	DEFECT_PRED_TOOL
Means of Verification:	Direct implementation on IDE, feature checklist, case-study
Dependency:	R-T2.3-3

ID:	R-T3.4-3
Type:	USABILITY
User story:	As an Operations Engineer I want to use the tool with a Command Line Interface
Requirement:	The defect-prediction tool could provide a command line interface.
Extended Description:	A command line interface will display information in a raw fashion which can be parsable by other tools and applications (e.g. presenting the results as json, xml, etc.).
Priority:	Could have
Affected Tools:	DEFECT_PRED_TOOL
Means of Verification:	Direct implementation on IDE, feature checklist, case-study

ID:	R-T3.4-4
Type:	USABILITY
User story:	As an Operations Engineer, I want to report missing defects and/or antipatterns to the tool
Requirement:	The defect-prediction tool could provide an interactive interface to allow developers and operators to report pieces of the infrastructure code where defects or antipatterns are present.
Extended Description:	An interactive interface allows users to report what's wrong in the infrastructure that has not been met by the tool. That way the tool can learn from those examples and improve its performance in defect prediction. At first glance, this step could be semi-automatic and thus involving the tool owner or tool management responsible to accept/reject user's reports
Priority:	Could have
Affected Tools:	DEFECT_PRED_TOOL
Means of Verification:	Direct implementation on IDE, feature checklist, case-study, user-study
Dependency:	R-T3.4-2 (uses)

ID:	R-T3.4-5
Type:	FUNCTIONAL_SUITABILITY
User story:	As a Operations Engineer/QoS Engineer/Release Manager I want the tool to show me a score of the correctness of my

	infrastructure's elements (e.g., a score of the defect threat-level)
Requirement:	The defect-prediction tool must provide a defect threat level to architecture elements and predict threat-level defects under certain infrastructure assumptions.
Extended Description:	The user must have the opportunity to prioritize the actions to solve defects.
Priority:	Must have
Affected Tools:	DEFECT_PRED_TOOL
Means of Verification:	Direct implementation on IDE, feature checklist, case-study
Dependency:	R-T3.4-1 (could use) R-T3.4-2 (could use) R-T3.4-3 (could use)
Conflict:	

ID:	R-T3.4-6
Type:	FUNCTIONAL_SUITABILITY
User story:	As an Operations Engineer I want the tool to be used automatically by my architecture's elements, such as an orchestrator
Requirement:	The defect-prediction tool must provide built in functionalities to be able to communicate with infrastructure elements in an automatic fashion.
Extended Description:	An orchestrator should be able to use this tool in an automatic manner to check deployment model for defects before starting the deployment
Priority:	Must have
Affected Tools:	DEFECT_PRED_TOOL
Means of Verification:	Direct implementation on IDE, feature checklist, case-study

ID:	R-T3.4-7
Type:	FUNCTIONAL_SUITABILITY
User story:	As an Operations Engineer I want to be able to avoid quickly any spelling mistakes or programming-language specific issues with language constructs
Requirement:	The defect-prediction tool must provide a linter to flag programming errors, bugs, style errors, and warnings.
Extended Description:	Code parsing before/upon deployment to detect and solve surface-level errors (simple anti patterns)
Priority:	Must have
Affected Tools:	DEFECT_PRED_TOOL
Means of Verification:	Direct implementation on IDE, feature checklist, case-study

ID:	R-T3.4-8
Type:	USABILITY
User story:	As an Operations Engineer I want to be able to find for specific defects
Requirement:	The defect-prediction tool must provide filters to decide which predefined defects to find
Extended Description:	The user should be able to filter which predefined defects are to be found
Priority:	Must have
Affected Tools:	DEFECT_PRED_TOOL
Means of Verification:	Direct implementation Feature checklist User study
Dependency:	R-T3.4-1 (could use) R-T3.4-2 (could use) R-T3.4-3 (could use)
Conflict:	

ID:	R-T3.4-9
Type:	FUNCTIONAL_SUITABILITY
User story:	As an Operations Engineer I want the tool to allow me to select the source (e.g., GitHub repository) from which the tool gathers data
Requirement:	The defect-prediction tool must be able to ingest data, also in real time, from multiple sources.
Extended Description:	The tool must provide data ingestion connectors to multiple sources (e.g., repositories like GitHub, Jira, etc.) to allow the users to link their repos to the tool. This allows the real-time data ingestion and defect prediction as well as to gather more data on which to constantly train the defect predictor model.
Priority:	Must have
Affected Tools:	DEFECT_PRED_TOOL
Means of Verification:	Direct implementation of connector to (at least) Github VCS Feature checklist
Dependency:	R-T3.4-10

ID:	R-T3.4-10
Type:	PERFORMANCE_EFFICIENCY
User story:	As an Operations Engineer I want the tool to find more defects than manual inspection

Requirement:	The defect-prediction tool must improve performances over manual inspection
Extended Description:	the tool needs to finish its analysis in less time with respect to manual inspection, without loss of effectiveness
Priority:	Must have
Affected Tools:	DEFECT_PRED_TOOL
Means of Verification:	User study with performance improved by at least 15%

ID:	R-T3.4-11
Type:	PERFORMANCE_EFFICIENCY
User story:	As an Operations Engineer I want the tool to decrease the time to inspect the infrastructure manually considerably
Requirement:	The tool must decrease bug-fixing times with respect to manual inspection
Extended Description:	The tool must diminish the time to detect defects with respect to manual inspection by at least 20%
Priority:	Must have
Affected Tools:	DEFECT_PRED_TOOL
Means of Verification:	User study with time improved by at least 20%

ID:	R-T3.3-1
Type:	FUNCTIONAL_SUITABILITY
User story:	As a QoS Engineer, I want to be able to validate the data pipelines which have been defined using a TOSCA models
Requirement:	The data pipeline testing module of the TESTING_TOOL must be able to test the quality and performance of data pipelines which are expressed using RADON models.
Extended Description:	The TESTING_TOOL must be able to utilise load testing to evaluate the performance of the input data pipeline, which are provided using RADON model templates.
Priority:	Must have
Affected Tools:	TESTING_TOOL
Means of Verification:	Tests with different RADON models
Dependency:	R-T5.4-1 (using the RADON Orchestrator data pipeline plugins to deploy and control data pipelines under testing)

ID:	R-T3.3-2
Type:	FUNCTIONAL_SUITABILITY
User story:	As a QoS Engineer, I want to be able to define custom data production profiles, based on which load testing data will be generated.
Requirement:	The data pipeline testing module of the TESTING_TOOL should support user configurable data production profiles.

Extended Description:	Data production profiles define what type and format data needs to be generated and to specify what is the profile of generated data flow.
Priority:	Should have
Affected Tools:	TESTING_TOOL
Means of Verification:	Using test case data production profiles for generating data into data pipeline and verifying that the produced data matches the profile.

ID:	R-T3.3-3
Type:	FUNCTIONAL_SUITABILITY
User story:	As a QoS Engineer, I want to be able to inject data producing components into the data pipeline (under testing) for generating test data into the pipeline.
Requirement:	The data pipeline testing module of the TESTING_TOOL must be able to inject additional pipeline components into the data pipeline for generating synthetic input data.
Extended Description:	The goal is to support creating and adding data pipeline components that mock the input streams by generating synthetic data into the data pipeline.
Priority:	Must have
Affected Tools:	TESTING_TOOL
Means of Verification:	Tests with different RADON data pipelines to verify that the components were injected and the modified data pipeline can be deployed successfully

ID:	R-T3.3-4
Type:	FUNCTIONAL_SUITABILITY
User story:	As a QoS Engineer, I want the testing tool to generate performance metrics for the data pipeline under testing.
Requirement:	The data pipeline testing module of the TESTING_TOOL should be able to ingest and analyze log data of the data pipeline under test to generate performance metrics for the deployed data pipeline.
Extended Description:	The goal is to parse and process the log data to generate performance metrics (e.g. latency/delay, error rate, dropped packets), which are required for evaluating the quality of the data pipeline.
Priority:	Should have
Affected Tools:	TESTING_TOOL
Means of Verification:	Tests with different data pipeline test cases to verify that the log data was parsed correctly.
Dependency:	R-T5.1-3

ID:	R-T3.3-5
Type:	USABILITY
User story:	As a QoS Engineer, I want to see the visualization of the computed performance metrics.
Requirement:	The data pipeline testing module of the TESTING_TOOL could be able to display graphical representation of performance metrics.
Extended Description:	The goal is to have visual graphs generated from the log data for each of the performance metrics.
Priority:	Could have
Affected Tools:	TESTING_TOOL
Means of Verification:	Tests with different data pipeline test cases to verify that graphs have been generated correctly

ID:	R-T3.3-6
Type:	FUNCTIONAL_SUITABILITY
User story:	As a QoS Engineer, I want to be able to configure critical lower and upper bounds of performance metrics and visualize them.
Requirement:	It could be useful if users can configure upper and lower bounds in the data pipeline testing module of the TESTING_TOOL for the performance metrics that are computed.
Extended Description:	Specifying upper and lower bounds for performance metrics could be used for defining visual alerts that indicate when the data pipeline under test is performing suboptimal to the desired performance.
Priority:	Could have
Affected Tools:	TESTING_TOOL
Means of Verification:	Tests with different data pipeline test cases and production profiles to verify that the tool recognises when upper and lower bounds of performance metrics have been reached

ID:	R-T3.3-7
Type:	USABILITY
User story:	As a QoS Engineer, I want to be able to use a graphical user interface for configuring the tool and displaying generated metrics and visualizations.
Requirement:	It could be useful for the data pipeline testing module of the TESTING_TOOL to have a graphical user interface for configuring tests and displaying test results.
Extended Description:	The goal is to provide a more user friendly interface for

	configuring tests and displaying results
Priority:	Could have
Affected Tools:	TESTING_TOOL
Means of Verification:	Tests with different data pipeline test cases to verify that all necessary information is displayed correctly on the user interface.

ID:	R-T3.3-8
Type:	USABILITY
User story:	As a QoS Engineer, I want to be able to configure a set of different data production profiles that will be checked/evaluated in a sequence in an automated manner.
Requirement:	It could be useful for the data pipeline testing module of the TESTING_TOOL to support running multiple different tests in a sequence on the same data pipeline.
Extended Description:	The goal is to support running a sequence of performance tests without requiring user intervention.
Priority:	Could have
Affected Tools:	TESTING_TOOL
Means of Verification:	Tests with a prepared set of testing configurations on different data pipelines to verify that all tests were run successfully
Dependency:	R-T5.4-1 (using the RADON Orchestrator data pipeline plugins to deploy and control data pipelines under testing)
Conflict:	

ID:	R-T.3.3-10
Type:	FUNCTIONAL_SUITABILITY
User story:	As a QoS Engineer, I want to be able to generate test cases from the test-annotated RADON models.
Requirement:	The FaaS testing module of the TESTING_TOOL must support the generation of test cases from RADON models that are augmented by the test-related annotations.
Extended Description:	The tool must be able to read and interpret the annotated RADON models, and to generate executable tests.
Priority:	Must have
Affected Tools:	TESTING_TOOL
Means of Verification:	Tests with RADON models, including synthetic ones and those from the case studies
Dependency:	RADON models must provide the respective annotations (R-T4.1-2, R-T4.1-3, R-T4.2-7,R-T4.3-8)

ID:	R-T.3.3-11
------------	------------

Type:	FUNCTIONAL_SUITABILITY
User story:	As a QoS Engineer, I want the CD pipeline to execute manually specified and generated tests.
Requirement:	The FaaS testing module of the TESTING_TOOL must support the execution of tests cases in the CD pipeline.
Extended Description:	The tool must be able to interpret and execute the test specifications. The execution involves potentially spawning parts of the testing environment.
Priority:	Must have
Affected Tools:	TESTING_TOOL
Means of Verification:	Tests with different test cases
Dependency:	- R-T.3.3-10 - Requirements for the CD pipeline for execution

ID:	R-T.3.3-12
Type:	FUNCTIONAL_SUITABILITY
User story:	As a QoS Engineer, I want my tests to be updated automatically based on production data.
Requirement:	The FaaS testing module of the TESTING_TOOL must be able to analyze monitoring data from production and update the annotations in the RADON model.
Extended Description:	The focus of the extraction is on the workload data but may also include inputs. Actually not the tests are updated but the RADON model annotations.
Priority:	Must have
Affected Tools:	TESTING_TOOL
Means of Verification:	Comparison of "original" measures with the results of running the extracted tests.
Dependency:	- RADON models must provide the respective annotations - The runtime platform must provide the respective data

ID:	R-T.3.3-13
Type:	FUNCTIONAL_SUITABILITY
User story:	As a QoS Engineer, I want the test results to be visualized.
Requirement:	The FaaS testing module of the TESTING_TOOL could have a report feature.
Extended Description:	The results might even be presented in the UI.
Priority:	Could have
Affected Tools:	TESTING_TOOL
Means of Verification:	Tests and demos
Dependency:	Possibly R-T4.3-7

ID:	R-T.3.3-14
Type:	USABILITY

User story:	As a QoS Engineer, I want to be able to specify tests using a GUI.
Requirement:	The FaaS testing module of the TESTING_TOOL must have a graphic user interface.
Extended Description:	The FaaS module of the TESTING_TOOL must graphically provide information about the generated tests, as well as the input data and state of the infrastructure, which will be used.
Priority:	Must have
Affected Tools:	TESTING_TOOL
Means of Verification:	Tests and demos

ID:	R-T.3.3-15
Type:	USABILITY
User story:	As a QoS Engineer, I want to be able to specify tests using CLI.
Requirement:	The FaaS testing module of the TESTING_TOOL should have a command line interface.
Extended Description:	A CLI can be used to specify test execution. Furthermore, a CLI can be useful to display, e.g., to display raw results, in a format parsable by other tools and applications (e.g., CSV, XML, JSON).
Priority:	Should have
Affected Tools:	TESTING_TOOL
Means of Verification:	Tests and demos

ID:	R-T.3.3-16
Type:	USABILITY
User story:	As a Software Engineer, I want continuous testing to be integrated into my development workflow
Requirement:	The FaaS testing module of the TESTING_TOOL must be integrated into DevOps practices.
Extended Description:	This includes DevOps properties such as multiple times, continuous development, fast feedback cycles, etc.
Priority:	Must have
Affected Tools:	TESTING_TOOL
Means of Verification:	Qualitative by demonstrating based on the RADON methodology and case studies

ID:	R-T3.2-1
Type:	FUNCTIONAL_SUITABILITY
User story:	As a Software Designer, I want to decompose the architecture of a monolithic application based on microservices.
Requirement:	Given a monolithic RADON model, the DECOMP_TOOL

	should be able to generate a coarse-grained RADON model.
Extended Description:	A RADON application or model may be said to be monolithic, coarse-grained or fine-grained depending on whether it consists of a monolith, microservices or serverless functions with separate storages.
Priority:	Should have
Affected Tools:	DECOMP_TOOL
Means of Verification:	Code review, unit tests and use cases
Dependency:	R-T4.2-1 to R-T4.2-3

ID:	R-T3.2-2
Type:	FUNCTIONAL_SUITABILITY
User story:	As a Software Designer, I want to decompose the architecture of a coarse-grained application based on serverless functions.
Requirement:	Given a coarse-grained RADON model, the DECOMP_TOOL should be able to generate a fine-grained RADON model.
Priority:	Should have
Affected Tools:	DECOMP_TOOL
Means of Verification:	Code review, unit tests and use cases
Dependency:	R-T4.2-1 to R-T4.2-3

ID:	R-T3.2-3
Type:	FUNCTIONAL_SUITABILITY
User story:	As an Operations Engineer, I want to optimize the deployment of an application on any cloud platform.
Requirement:	Given a platform-independent RADON model, the DECOMP_TOOL must be able to obtain an optimal deployment scheme that minimizes the operating costs on a specific cloud platform under the performance requirements.
Extended Description:	A platform-independent RADON model comprises nodes and relationships that are agnostic of any cloud platform.
Priority:	Must have
Affected Tools:	DECOMP_TOOL
Means of Verification:	Code review, unit tests and use cases
Dependency:	R-T4.2-1 to R-T4.2-3

ID:	R-T3.2-4
Type:	FUNCTIONAL_SUITABILITY
User story:	As an Operations Engineer, I want to optimize the deployment of an application on a particular cloud platform.
Requirement:	Given a platform-specific RADON model, the DECOMP_TOOL must be able to obtain an optimal

	deployment scheme that minimizes the operating costs on the target cloud platform under the performance requirements.
Extended Description:	A platform-specific RADON model comprises nodes and relationships that are dependent on a particular cloud platform.
Priority:	Must have
Affected Tools:	DECOMP_TOOL
Means of Verification:	Code review, unit tests and use cases
Dependency:	R-T4.2-1 to R-T4.2-3

ID:	R-T3.2-5
Type:	FUNCTIONAL_SUITABILITY
User story:	As an Operations Engineer, I want to enhance the accuracy of the description for an application.
Requirement:	Given a deployable RADON model, the DECOMP_TOOL could be able to refine certain properties of the nodes and relationships using runtime monitoring data.
Extended Description:	The refined properties may include arrival rates, numbers of users, think times, network latency and service times.
Priority:	Could have
Affected Tools:	DECOMP_TOOL
Means of Verification:	Code review, unit tests and use cases
Dependency:	R-T4.2-1 to R-T4.2-3

ID:	R-T3.2-6
Type:	FUNCTIONAL_SUITABILITY
User story:	As a Software Designer, I want to add new features to a monolithic application as microservices or serverless functions.
Requirement:	The DECOMP_TOOL should be able to carry out architecture decomposition, deployment optimization and accuracy enhancement for a mixed-grained RADON model.
Extended Description:	A RADON application or model is said to be mixed-grained if it consists of a monolith, microservices or serverless functions with separate storages.
Priority:	Should have
Affected Tools:	DECOMP_TOOL
Means of Verification:	Code review, unit tests and use cases
Dependency:	R-T4.2-1 to R-T4.2-3

ID:	R-T3.2-7
Type:	FUNCTIONAL_SUITABILITY
User story:	As a Software Designer, I want to integrate an application

	with heterogeneous cloud technologies.
Requirement:	The DECOMP_TOOL should be able to carry out architecture decomposition, deployment optimization and accuracy enhancement for a RADON model with heterogeneous cloud technologies.
Extended Description:	The supported technologies may include containers, virtual machines, object storages, block storages, databases, message queues, data streams and in-memory caches.
Priority:	Should have
Affected Tools:	DECOMP_TOOL
Means of Verification:	Code review, unit tests and use cases
Dependency:	R-T4.2-1 to R-T4.2-3

ID:	R-T3.2-8
Type:	FUNCTIONAL_SUITABILITY
User story:	As a Software Designer, I want to distribute an application across multiple cloud platforms.
Requirement:	The DECOMP_TOOL should be able to carry out architecture decomposition, deployment optimization and accuracy enhancement for a RADON model across multiple cloud platforms.
Extended Description:	The supported platforms may include Amazon Web Services, Google Cloud and Microsoft Azure.
Priority:	Should have
Affected Tools:	DECOMP_TOOL
Means of Verification:	Code review, unit tests and use cases
Dependency:	R-T4.2-1 to R-T4.2-3

ID:	R-T3.2-9
Type:	USABILITY
User story:	As an Operations Engineer, I want to decompose the architecture of an application at different granularity levels.
Requirement:	The DECOMP_TOOL should be able to allow the option of specifying the granularity level for architecture decomposition and generate a grained RADON model at that level.
Extended Description:	The available levels may include coarse-grained, fine-grained and mixed-grained.
Priority:	Should have
Affected Tools:	DECOMP_TOOL
Means of Verification:	Code review, unit tests and use cases
Dependency:	R-T4.2-1 to R-T4.2-3

ID:	R-T3.2-10
------------	-----------

Type:	USABILITY
User story:	As an Operations Engineer, I want to optimize the deployment of an application with different solution methods.
Requirement:	The DECOMP_TOOL should be able to allow the option of specifying the solution method for deployment optimization and obtain the optimal deployment scheme with that method.
Extended Description:	The available methods may include cutting planes, branch & bound and generic heuristics.
Priority:	Should have
Affected Tools:	DECOMP_TOOL
Means of Verification:	Code review, unit tests and use cases
Dependency:	R-T4.2-1 to R-T4.2-3

ID:	R-T3.2-11
Type:	USABILITY
User story:	As an Operations Engineer, I want to optimize the deployment of an application in a definite time.
Requirement:	The DECOMP_TOOL should be able to allow the option of specifying the time limit for deployment optimization and return a sub-optimal deployment scheme upon timeout.
Extended Description:	A sub-optimal deployment scheme is the best among those found feasible so far.
Priority:	Should have
Affected Tools:	DECOMP_TOOL
Means of Verification:	Code review, unit tests and use cases
Dependency:	R-T4.2-1 to R-T4.2-3

ID:	R-T3.2-12
Type:	FUNCTIONAL_SUITABILITY
User story:	As a Software Designer/QoS Engineer/Operations Engineer, I want to be able to search for the best RADON model to fit a set of soft constraints expressed in the CDL.
Requirement:	Given a space of possible RADON models, the tool could compute an optimal RADON model with respect to CDL constraints. The computation for this may take place offline.
Extended Description:	-- Any returned model would comply with the hard constraints. -- Optimality would be defined in terms of the soft constraints.
Priority:	Could have
Affected Tools:	CDL
Means of Verification:	Code review, unit tests and use cases
Dependency:	R-T4.1-1 to R-T4.1-4

ID:	R-T3.2-13
Type:	FUNCTIONAL_SUITABILITY
User story:	As a Software Designer/QoS Engineer/Operations Engineer, I want to be able to use the CDL tool to get suggestions of how to improve my RADON model, which already complies with my constraints, but is not necessarily optimal wrt my soft constraints.
Requirement:	Given a compliant sub-optimal RADON model, the tool could provide suggestions, which would improve its score with respect to the CDL soft constraints, while keeping the change to the original RADON model as small as possible. This computation should be fast enough to be used by a user interactively.
Extended Description:	By compliant sub-optimal RADON model, we mean one that complies with the hard constraints, but does not score as well as it could with respect to the soft constraints.
Priority:	Could have
Affected Tools:	CDL
Means of Verification:	Code review, unit tests and use cases
Dependency:	R-T4.1-1 to R-T4.1-4

4. WP4: Modelling Environment Requirements

ID:	R-T4.1-1
Type:	FUNCTIONAL_SUITABILITY
User story:	As a QoS Engineer/Software Designer I want to express pre/post conditions for serverless functions using a CDL.
Requirement:	The CDL must be able to express pre/post conditions of serverless functions regarding security/performance.
Extended Description:	The CDL must provide standard mechanisms to express pre/post conditions for serverless functions.
Priority:	Must have
Affected Tools:	CDL
Means of Verification:	A check that the CDL can express pre/post conditions for a set of different serverless functions.

ID:	R-T4.1-2
Type:	FUNCTIONAL_SUITABILITY
User story:	As a QoS Engineer/Software Designer I want to express hard constraints on security and performance of serverless functions, microservices and data pipelines using the CDL.
Requirement:	The CDL must be able to express hard constraints on the required security/performance.

Extended Description:	Security conditions include constraints on the encryption / access control of personal data. One example of a performance constraint is a guarantee of low-latency.
Priority:	Must have
Affected Tools:	CDL
Means of Verification:	A check that the CDL can express a set of required hard constraints for a set of different use cases.

ID:	R-T4.1-3
Type:	FUNCTIONAL_SUITABILITY
User story:	As a QoS Engineer/Software Designer I want to express that a set of nodes respect a given architectural pattern (e.g. a Lambda).
Requirement:	The CDL must be able to express hard constraints on the architectural patterns of sets of nodes.
Extended Description:	The CDL must provide standard mechanisms to express hard constraints on the architectural patterns of sets of nodes.
Priority:	Must have
Affected Tools:	CDL
Means of Verification:	A check that the CDL can express at least three different architectural patterns using simple "built in" predicates, and that the user is able to define these same architectural patterns manually, without using the "built-ins".

ID:	R-T4.1-4
Type:	FUNCTIONAL_SUITABILITY
User story:	As a QoS Engineer/Software Designer I want to express soft constraints on security and performance of serverless functions, microservices and data pipelines using the CDL.
Requirement:	The CDL should be able to express soft constraints on the required security/performance.
Extended Description:	For example, there may be a hard constraint on the maximum latency, but a soft constraint expressing that even if the latency is within the required level, the lowest possible latency is preferred. The soft constraints should be weighted and prioritised, in order to allow us to reason about trade-offs (e.g. in some cases, soft constraints on latency may be less important than soft constraints on security).
Priority:	Should have
Affected Tools:	CDL
Means of Verification:	A check that the CDL can express a set of required soft constraints for a set of different use cases.

ID:	R-T4.1-5
------------	----------

Type:	FUNCTIONAL_SUITABILITY
User story:	As a QoS Engineer/Operations Engineer I want to verify if specified constraints are satisfied for a given RADON model.
Requirement:	The verification tool must be able to check that hard constraints are guaranteed to be satisfied by a given RADON model.
Extended Description:	In case conditions are not satisfied, the tool should be able to provide a counter example as an explanation, detailing which nodes violate which constraint. For some hard constraints it may be necessary to give a search space (e.g. a maximum bound on the time steps).
Priority:	Must have
Affected Tools:	VERIFIC_TOOL
Means of Verification:	Formal proof of the soundness and completeness of the verification algorithm.
Dependency:	R-T4.1-1, R-T4.1-2, R-T4.1-3

ID:	R-T4.1-6
Type:	PERFORMANCE_EFFICIENCY
User story:	As a QoS Engineer/Operations Engineer I want to verify if specified constraints are satisfied for a given RADON model in real time.
Requirement:	This computation to check whether hard constraints are satisfied by a RADON model should return within a predefined maximum time.
Extended Description:	The verification tool must provide standard mechanisms to verify if the hard constraints specified for a given RADON model are not violated.
Priority:	Must have
Affected Tools:	VERIFIC_TOOL
Means of Verification:	Emperical evaluation on a test set of large RADON models.
Dependency:	R-T4.1-5

ID:	R-T4.1-7
Type:	FUNCTIONAL_SUITABILITY
User story:	As a QoS Engineer/Operations Engineer I want to test RADON model for existence of race conditions and/or execution loops.
Requirement:	The verification tool should be able to check for the existence of potential race conditions and/or execution loops and related deadlocks that could happen.
Extended Description:	For instance different operators receiving requests for different actions regarding the same patient or which require sharing of resources of robotic assistants. In the case that

	such events can occur, the tool should provide an example trace as explanation.
Priority:	Should have
Affected Tools:	VERIFIC_TOOL
Means of Verification:	Formal proof of the soundness and completeness of the verification algorithm.

ID:	R-T4.1-8
Type:	PERFORMANCE_EFFICIENCY
User story:	As a QoS Engineer/Operations Engineer I want to test RADON model for existence of race conditions and/or execution loops in real time.
Requirement:	The computation to check for race conditions, execution loops and deadlocks should return within a predefined maximum time.
Extended Description:	The verification tool must provide standard mechanisms to test a given RADON model for existence of race conditions and/or execution loops.
Priority:	Should have
Affected Tools:	VERIFIC_TOOL
Means of Verification:	Empirical evaluation on a test set of large RADON models, some with race conditions, loops and/or deadlocks, some without.
Dependency:	R-T4.1-7

ID:	R-T4.1-9
Type:	FUNCTIONAL_SUITABILITY
User story:	As a QoS Engineer/Operations Engineer, I would like to be able to use the verification tool to get a suggestion of how to correct a RADON model that violates some of my constraints.
Requirement:	Given a RADON model that violates some constraints, the tool could provide corrections to the RADON model to ensure that it complies with the constraints.
Extended Description:	We may search for minimal changes. Minimal changes are desirable to keep the model as close to the model that the user intended.
Priority:	Could have
Affected Tools:	VERIFIC_TOOL
Means of Verification:	Formal proofs of completeness (the algorithm finds a correction if one exists), soundness (any correction reported by the algorithm is guaranteed to be valid).
Dependency:	R-T4.1-1, R-T4.1-2, R-T4.1-3

ID:	R-T4.1-10
------------	-----------

Type:	PERFORMANCE_EFFICIENCY
User story:	As a QoS Engineer/Operations Engineer, I would like the suggestion of how to correct a RADON model that violates some of my constraints to be provided in real time.
Requirement:	This computation should complete within a predefined maximum time.
Extended Description:	After the constraint violation has been identified, the verification tool could recommend the ways to correct the given model.
Priority:	Could have
Affected Tools:	VERIFIC_TOOL
Means of Verification:	Empirical evaluation on a test set of large violating RADON models.
Dependency:	R-T4.1-9

ID:	R-T4.1-11
Type:	FUNCTIONAL_SUITABILITY
User story:	As a QoS Engineer/Operations Engineer, I want to be able to guarantee that in a dynamic situation, where new devices are being continually added and taken away, these dynamic changes are guaranteed not to cause my RADON model to violate my hard constraint
Requirement:	Given a space of possible RADON models (in a dynamic situation, new devices such as robotic assistants may be added/taken away), the tool could verify that any RADON model in the space complies with a set of hard constraints.
Extended Description:	This computation may take place offline.
Priority:	Could have
Affected Tools:	VERIFIC_TOOL
Means of Verification:	Proof of soundness and completeness. This constitutes proving that any violating RADON model within the space will be found (completeness) and that if a RADON model in the space is reported to be violating, it does indeed violate at least one hard constraint (soundness).
Dependency:	R-T4.1-1, R-T4.1-2, R-T4.1-3

ID:	R-T4.2-1
Type:	FUNCTIONAL_SUITABILITY
User story:	As a Software Designer/Software Developer/Operations Engineer, I want to model heterogeneous application topologies (FaaS, microservices, third-party and cloud-native resources).
Requirement:	The models must allow expressing combinations of different deployment types including paradigm-specific elements, e.g., events and triggers.

Extended Description:	RADON model must allow expressing combinations of different deployment types including paradigm-specific elements, e.g., events and triggers.
Priority:	Must have
Affected Tools:	GRAPHMODEL_TOOL
Means of Verification:	Model review with partners and case study.

ID:	R-T4.2-2
Type:	MAINTAINABILITY
User story:	As a Software Designer/Software Developer/Operations Engineer, I want to use a library/repository of existing blueprints to model desired applications.
Requirement:	In RADON we should provide a repository (e.g., GitHub) to provide reusable types and blueprints.
Extended Description:	To simplify modeling of complex topologies, existing and deployable blueprints must be available as building blocks for reuse.
Priority:	Should have
Affected Tools:	GRAPHMODEL_TOOL
Means of Verification:	Case study.
Dependency:	T5.2

ID:	R-T4.2-3
Type:	FUNCTIONAL_SUITABILITY
User story:	As a Software Designer/Software Developer/Operations Engineer, I want to represent behavioral aspects (e.g. data pipelines, event flow) in my application models.
Requirement:	The models must be able to define different kinds of data processing tasks and control flow elements in order to express the behavior of my application.
Extended Description:	For example, jobs that process analytics data or moving files between different storage systems. Further, the modeling of timed schedules must be supported.
Priority:	Must have
Affected Tools:	GRAPHMODEL_TOOL
Means of Verification:	Model review with partners and case study.
Dependency:	T5.4

ID:	R-T4.2-4
Type:	FUNCTIONAL_SUITABILITY
User story:	As a Software Designer/Software Developer/QoS Engineer, I want to define preconditions for data pipelines in the application model.
Requirement:	The models should be able to define certain preconditions

	for filtering which data objects to move/stream through the pipeline.
Extended Description:	For example to filter which types of log entries should be ingested or to only move files that are older than X hours.
Priority:	Should have
Affected Tools:	GRAPHMODEL_TOOL
Means of Verification:	Model review with partners and case studies.
Dependency:	T5.4

ID:	R-T4.2-5
Type:	FUNCTIONAL_SUITABILITY
User story:	As a Software Designer/Software Developer/QoS Engineer, I want to express scaling attributes for data processing components in my application models.
Requirement:	The models should be able to define how and when to scale certain computing resources.
Extended Description:	For example, the computing resources allocated to data pipelines in order to be able to respond to the changes in the rate of ingested data.
Priority:	Should have
Affected Tools:	GRAPHMODEL_TOOL
Means of Verification:	Model review with partners and case study.
Dependency:	T5.1, T5.4

ID:	R-T4.2-6
Type:	FUNCTIONAL_SUITABILITY
User story:	As a Software Designer/Software Developer/QoS Engineer, I can define data (un-)compression for data pipelines.
Requirement:	The models could define configurations regarding data compression and uncompression for certain processing components.
Extended Description:	For example to reduce the size of data that is flown through the specified data pipeline.
Priority:	Could have
Affected Tools:	GRAPHMODEL_TOOL
Means of Verification:	Model review with partners and case study.
Dependency:	T5.4

ID:	R-T4.2-7
Type:	FUNCTIONAL_SUITABILITY
User story:	As a QoS Engineer/Software Developer, I want to define test case specifications in the models.
Requirement:	The models must be able to include the description of test cases for certain components (annotate test-related

	information).
Extended Description:	There must be test case specifications for components in the model, which are used to generate tests in the final runtime.
Priority:	Must have
Affected Tools:	GRAPHMODEL_TOOL
Means of Verification:	Model review with partners and case study.

ID:	R-T4.3-1
Type:	FUNCTIONAL_SUITABILITY
User story:	As an Operations Engineer, I want to create new application blueprints using a graphical modeling tool.
Requirement:	The graphical modeling tool (GMT, Winery) must provide a GUI which is integrated into the IDE.
Extended Description:	For example the GUI is launched by clicking on kind of a <new blueprint> action in the IDE.
Priority:	Must have
Affected Tools:	GRAPHMODEL_TOOL
Means of Verification:	Manual testing in the IDE based on case studies from T4.2.
Dependency:	T2.3

ID:	R-T4.3-2
Type:	FUNCTIONAL_SUITABILITY
User story:	As a Software Developer, I want to be able to navigate to the source code implementing the business logic of the respective component in the model.
Requirement:	In the GMT, it must be possible to navigate to the respective workspace where the source code is maintained.
Extended Description:	By clicking on a modeled component a user must be able to navigate to the actual source code of the component.
Priority:	Must have
Affected Tools:	IDE
Means of Verification:	Manual testing in the IDE based on case studies from T4.2.
Dependency:	T2.3

ID:	R-T4.3-3
Type:	FUNCTIONAL_SUITABILITY
User story:	As an Operations Engineer, I want to navigate to the source code implementing the deployment logic of the respective component in the model.
Requirement:	In the GMT, it should be possible to navigate to a workspace where the deployment logic is located.
Extended Description:	It should be possible to switch from the deployment logic view of the chosen topology node in GMT to the corresponding source code in IDE.

Priority:	Should have
Affected Tools:	IDE
Means of Verification:	Manual testing in the IDE based on case studies from T4.2.
Dependency:	T2.3

ID:	R-T4.3-4
Type:	FUNCTIONAL_SUITABILITY
User story:	As a QoS Engineer/Software Designer, I want to annotate blueprint components with modeled constraints.
Requirement:	In the GMT, a user must be able to define a set of constraints (based on the CDL) for one or more components.
Extended Description:	Assuming there are a set of predefined constraints, a user must be able to use them and relate them to the corresponding components. It would be nice to have a graphical approach for common (built in) constraints, but we must allow the user to open the text editor to allow them to be more flexible.
Priority:	Must have
Affected Tools:	GRAPHMODEL_TOOL
Means of Verification:	Unit testing, manual testing in the IDE based on case studies from T4.2.

ID:	R-T4.3-5
Type:	MAINTAINABILITY
User story:	As a Software Designer/Software Developer/Operations Engineer, I want to import existing blueprints for reuse or modification.
Requirement:	A user could be able to import existing models that can then be reused when creating new ones.
Extended Description:	This could be done through the IDE or directly in the modeling tool.
Priority:	Could have
Affected Tools:	GRAPHMODEL_TOOL
Means of Verification:	Unit testing, manual testing in the IDE based on case studies from T4.2.
Dependency:	T5.2

ID:	R-T4.3-6
Type:	COMPATIBILITY
User story:	As a Software Designer/Software Developer/Operations Engineer, I want my modeling tool to support integrations to other ones.
Requirement:	A user should be able to trigger certain tools from the modeling tool.
Extended Description:	For example to trigger the decomposition tool or the

	verification tool.
Priority:	Should have
Affected Tools:	GRAPHMODEL_TOOL
Means of Verification:	Manual testing in the IDE.
Dependency:	T4.1; T3.2; T3.3

ID:	R-T4.3-7
Type:	FUNCTIONAL_SUITABILITY
User story:	As a QoS Engineer/Software Designer, if my RADON model violates hard constraints in the CDL I would like to see a graphical explanation of which constraints are violated and by which parts of the RADON model
Requirement:	Given a RADON model which does not comply with a set of hard constraints, the graphical modelling tool should be able to graphically represent the explanation generated by the verification tool.
Extended Description:	For example highlighting which nodes are involved in the violation and which constraints are violated.
Priority:	Should have
Affected Tools:	GRAPHMODEL_TOOL
Means of Verification:	Manual testing in the IDE.
Dependency:	T4.1

ID:	R-T4.3-8
Type:	FUNCTIONAL_SUITABILITY
User story:	As a QoS Engineer/Software Developer, I want to annotate blueprints with test-related information.
Requirement:	In the GMT, it must be possible to use predefined or to create new test case specifications that a user can use to annotate modeled components.
Extended Description:	To support continuous testing of modelled cloud applications, the GMT must support specification of test-related information using a graphical user interface.
Priority:	Must have
Affected Tools:	GRAPHMODEL_TOOL
Means of Verification:	Unit testing, manual testing in the IDE based on case studies from T4.2.

ID:	R-T4.3-9
Type:	PERFORMANCE_EFFICIENCY
User story:	As a Software Designer/Software Developer/Operations Engineer, I want to model tens or hundreds of elements without any serious performance impact.
Requirement:	In the GMT, it must be possible to model an amount of up to

	two hundred elements (i.e., nodes, relations).
Extended Description:	When modeling functions, events, and triggers it is most probably required to model a bunch of elements (tens or event hundreds). This requirement makes sure that a certain degree of performance is guaranteed when using the modeling tool.
Priority:	Must have
Affected Tools:	GRAPHMODEL_TOOL
Means of Verification:	Manual testing using serverless application deployments having 100 functions with one event trigger per function.

ID:	R-T4.3-10
Type:	PERFORMANCE_EFFICIENCY
User story:	As a Software Designer/Software Developer/Operations Engineer, I want to have the possibility to handle the visual complexity of a huge amount of FaaS elements.
Requirement:	The GMT could provide a feature to group or abstract certain elements in order to reduce the visual complexity of tens or hundreds of FaaS components.
Extended Description:	Pure FaaS applications mostly consist of tens or hundreds of elements and components, such as functions, events, and used cloud services. This could result in a visual overload of elements on the drawing canvas of the GMT.
Priority:	Could have
Affected Tools:	GRAPHMODEL_TOOL
Means of Verification:	Manual testing using serverless application deployments having 100 functions with one event trigger per function.

ID:	R-T4.4-1
Type:	FUNCTIONAL_SUITABILITY
User story:	As a Software Designer/Software Developer/Operations Engineer, I want to export available blueprints in an orchestrator supported format.
Requirement:	The bundle which is exported from the modeling tool must be processable by the RADON orchestrator.
Extended Description:	A user must be able to export the application model (including all required artifacts, such as business logic and deployment logic artifacts) to a portable archive.
Priority:	Must have
Affected Tools:	GRAPHMODEL_TOOL
Means of Verification:	Direct testing using xOpera based on the case studies from T4.2.
Dependency:	T5.1/ORCHESTRATOR, DELIVERY_TOOLCHAIN

ID:	R-T4.4-2
------------	----------

Type:	FUNCTIONAL_SUITABILITY
User story:	As a Operations Engineer/Release Manager, I want to have support for different orchestration engines/tools.
Requirement:	The GMT could provide an option to export a blueprint in different formats to use other orchestration tools, such as OpenTOSCA or Terraform.
Extended Description:	To support applications modeller with additional options for orchestrating the deployment of modelled cloud applications, GMT could support export into different deployment model formats.
Priority:	Could have
Affected Tools:	GRAPHMODEL_TOOL
Means of Verification:	Direct testing using xOpera based on the case studies from T4.2.
Dependency:	T5.1/ ORCHESTRATOR, DELIVERY_TOOLCHAIN

ID:	R-T4.4-3
Type:	FUNCTIONAL_SUITABILITY
User story:	As a Software Designer/Software Developer/Operations Engineer, I want to reflect/import the output of other RADON tools inside the GMT.
Requirement:	The GMT could provide the possibility to import different output formats produced by the integrated RADON tools.
Extended Description:	As many tools work with RADON models, it could be the case that Winery need to consume different formats.
Priority:	Could have
Affected Tools:	GRAPHMODEL_TOOL
Means of Verification:	Manual testing.
Dependency:	T4.1; T3.2; T3.3

5. WP5: Runtime Environment Requirements

ID:	R-T5.1-1
Type:	FUNCTIONAL_SUITABILITY
User story:	As an Operations Engineer, I want to use orchestrator through command line
Requirement:	The orchestrator tasks must be executable through CLI
Extended Description:	Orchestrator should provide CLI to manage the deployment of blueprints on serverless and FaaS providers
Priority:	Should have
Affected Tools:	ORCHESTRATOR
Means of Verification:	Test if deployment of RADON model is possible using only CLI.

ID:	R-T5.1-2
------------	----------

Type:	FUNCTIONAL_SUITABILITY
User story:	As a Software Developer, I want to describe application in TOSCA YAML 1.2
Requirement:	A Software Developer describes the application architecture and dependencies at least in TOSCA YAML 1.2
Extended Description:	The TOSCA blueprints, input to orchestrator, must be written in TOSCA YAML 1.2 syntax, at minimum.
Priority:	Must have
Affected Tools:	DELIVERY_TOOLCHAIN
Means of Verification:	The orchestrator should report error in case the blueprint is not in TOSCA YAML 1.2

ID:	R-T5.1-3
Type:	USABILITY
User story:	As an Operations Engineer, I want to see the status of runtime deployment of application in each stage
Requirement:	The runtime toolchain need to provide a status on each stage of deployment of the application on the underlying architecture
Extended Description:	Using different technologies for monitoring the deployment process, a service or tool must be developed to keep users updated on the status of the deployment process. In case of errors they should be immediatly reported.
Priority:	Must have
Affected Tools:	MONITORING_SYSTEM
Means of Verification:	During runtime execution the log of activites/steps must be available.

ID:	R-T5.1-4
Type:	RELIABILITY
User story:	As an Operations Engineer, I want to be able to validate the deployment of application
Requirement:	At the end of deployment the deployment of services needs to be verified and its dependencies
Extended Description:	Verification methodology must be developed to verify the correctness of deployment application and if its working as expected.
Priority:	Must have
Affected Tools:	TESTING_TOOL
Means of Verification:	Integration tests could be developed to automatically test the deployment of application and if all rules defined in blueprint were correctly applied

ID:	R-T5.1-5
Type:	PERFORMANCE_EFFICIENCY

User story:	As an Operations Engineer, I want to update my deployment based on a small change in the model.
Requirement:	The orchestrator should be able to calculate the diff between the current state and the desired state expressed by a new model version and redeploy only the difference.
Extended Description:	The orchestrator only need to execute delta updates instead of redeploying all services, making redeployment more efficient and in shorter time than with conventional way of deleting everything and deploy it again.
Priority:	Must have
Affected Tools:	ORCHESTRATOR
Means of Verification:	Manually check if only the changed part of blueprint was deployed

ID:	R-T5.1-6
Type:	PORTABILITY
User story:	As an Operations Engineer, I want to be able to deploy TOSCA blueprint to OpenFaas
Requirement:	Support of FaaS deployment to OpenFaas
Extended Description:	The TOSCA bluprints and Ansible playbooks should be developed to deploy RADON model on OpenFaaS
Priority:	Must have
Affected Tools:	TEMPLATE_LIBRARY
Means of Verification:	Test if such playbooks and blueprints are available in Template Library repository and its possible to deploy toy example to OpenFaas platform only using playbooks and blueprints from Template Library.

ID:	R-T5.1-7
Type:	PORTABILITY
User story:	As an Operations Engineer, I want to be able to deploy TOSCA blueprint to AWS
Requirement:	Support of FaaS deployment to AWS cloud platform
Extended Description:	The TOSCA bluprints and Ansible playbooks should be developed to deploy RADON model on AWS cloud platform
Priority:	Must have
Affected Tools:	TEMPLATE_LIBRARY
Means of Verification:	Test if such playbooks and blueprints are available in Template Library repository and its possible to deploy toy example to AWS cloud platform only using playbooks and blueprints from Template Library.

ID:	R-T5.2-8
Type:	PORTABILITY



Deliverable 2.1: Initial requirements and baselines

User story:	As an Operations Engineer, I want to be able to deploy TOSCA blueprint to Google
Requirement:	Support of FaaS deployment to Google cloud platform
Extended Description:	The TOSCA blueprints and Ansible playbooks should be developed to deploy RADON model on Google Cloud
Priority:	Could have
Affected Tools:	TEMPLATE_LIBRARY
Means of Verification:	Test if such playbooks and blueprints are available in Template Library repository and the possibility to deploy toy example to Google Cloud platform only using playbooks and blueprints from Template Library.

ID:	R-T5.2-9
Type:	PORTABILITY
User story:	As an Operations Engineer, I want to be able to deploy TOSCA blueprint to Azure
Requirement:	Support of FaaS deployment to Azure cloud platform
Extended Description:	The TOSCA blueprints and Ansible playbooks should be developed to deploy RADON model on Azure
Priority:	Must have
Affected Tools:	TEMPLATE_LIBRARY
Means of Verification:	Test if such playbooks and blueprints are available in Template Library repository and its possible to deploy toy example to Google Cloud platform only using playbooks and blueprints from Template Library.

ID:	R-T5.2-10
Type:	PORTABILITY
User story:	As an Operations Engineer, I want to be able to deploy my application to regular VMs
Requirement:	Support deployment to regular VMs
Extended Description:	The TOSCA blueprints and Ansible playbooks should be developed to deploy RADON model on regular VMs
Priority:	Must have
Affected Tools:	TEMPLATE_LIBRARY
Means of Verification:	Test if such playbooks and blueprints are available in Template Library repository and its possible to deploy toy example to regular VMs.

ID:	R-T5.2-11
Type:	PORTABILITY
User story:	As an Operations Engineer, I want to be able to deploy my containerized application to microservices architecture
Requirement:	Support deployment to microservices architecture

Extended Description:	The TOSCA blueprints and Ansible playbooks should be developed to deploy RADON model on microservices architecture using Docker or Kubernetes container technologies
Priority:	Must have
Affected Tools:	TEMPLATE_LIBRARY
Means of Verification:	Test if such playbooks and blueprints are available in Template Library repository and its possible to deploy toy example to microservices architecture.

ID:	R-T5.4-1
Type:	FUNCTIONAL_SUITABILITY
User story:	As an Operations Engineer, I want to be able to deploy data pipelines and control their life-cycle using RADON toolchain
Requirement:	The data pipeline module of the Orchestrator must be able to orchestrate data pipelines
Extended Description:	Orchestrator must provide support for orchestrating data pipelines.
Priority:	Must have
Affected Tools:	ORCHESTRATOR
Means of Verification:	Evaluate whether the tool is able to successfully deploy data pipelines which have been defined using RADON models.

ID:	R-T5.4-2
Type:	FUNCTIONAL_SUITABILITY
User story:	As an Operations Engineer, I want to be able to schedule data pipelines periodically
Requirement:	The data pipeline module of the Orchestrator must support cron based scheduled data pipelines
Extended Description:	Goal is to allow users to configure specific times and intervals when the data pipeline is executed.
Priority:	Must have
Affected Tools:	ORCHESTRATOR, DELIVERY_TOOLCHAIN
Means of Verification:	Test data pipeline execution with different time based periodic schedules and verify that pipeline is executed every single time.

ID:	R-T5.4-3
Type:	FUNCTIONAL_SUITABILITY
User story:	As an Operations Engineer, I want to be able to trigger the data pipeline execution in Event-driven manner
Requirement:	The data pipeline module of the Orchestrator should support event based scheduled data pipelines

Extended Description:	Goal is to support data pipelines that are executed on demand or in an event-driven manner.
Priority:	Should have
Affected Tools:	ORCHESTRATOR
Means of Verification:	Test data pipeline execution with different event based triggers and verify that pipeline is executed every single time

ID:	R-T5.4-4
Type:	RELIABILITY
User story:	As an QoS Engineer, I want to be notified if the data pipeline execution fails
Requirement:	It would be useful for the data pipeline module of the Orchestrator to support logging and generating alerts on pipeline task failures.
Extended Description:	Goal is to support notifications for more critical data pipelines, which would be raised when data pipeline tasks fail or are overloaded.
Priority:	Should have
Affected Tools:	ORCHESTRATOR
Means of Verification:	Test data pipelines that have notifications enabled, force critical situations to happen (e.g. by sending malformed data, shutting down specific services or generating too much data) and verify that alarms are raised

ID:	R-T5.4-5
Type:	FUNCTIONAL_SUITABILITY
User story:	As an Operations Engineer, I want to be able to deploy data pipelines across multiple (cloud) systems
Requirement:	The data pipeline module of the Orchestrator should support deployment of data pipelines which automate movement of data between two or more clouds
Extended Description:	Goal is to support data pipelines that migrate data between different cloud providers and availability zones.
Priority:	Should have
Affected Tools:	ORCHESTRATOR
Means of Verification:	Test data pipeline orchestration where different parts of the pipeline are deployed on different clouds

ID:	R-T5.4-6
Type:	FUNCTIONAL_SUITABILITY
User story:	As an Operations Engineer, I want to be able to deploy data pipelines that trigger NLP data analytics jobs
Requirement:	The data pipeline module must support data pipelines tasks that initiate data analytics tasks for processing data moving

	through the pipeline
Extended Description:	Goal is to support defining data pipelines that initiate analytics jobs in large scale data processing frameworks (e.g Hadoop, Spark, Flink) as one of the pipeline actions.
Priority:	Must have
Affected Tools:	ORCHESTRATOR
Means of Verification:	Test data pipelines that include initiating data analytics jobs as one or more pipeline tasks and verify that resulting data is correct.

ID:	R-T5.4-7
Type:	SECURITY
User story:	As an Operations Engineer, I want to be able to configure encryption for data moving through the data pipeline
Requirement:	The data pipeline module of the Orchestrator should support configuring encryption between data pipeline tasks when data needs to be moved between systems
Extended Description:	Goal is to support automatic data encryption pipes when data is transported between/outside cloud environment. Service mesh can be adapted for this.
Priority:	Should have
Affected Tools:	ORCHESTRATOR
Means of Verification:	Test data pipelines that move data between two systems. Analyze network traffic and verify data is not sent unencrypted.

ID:	R-T5.4-8
Type:	PORTABILITY
User story:	As an Operations Engineer, I want to deploy data pipelines to AWS data pipeline service
Requirement:	The data pipeline module of the Orchestrator should support deploying data pipelines expressed using TOSCA models into the AWS data pipeline service.
Extended Description:	The goal is to support deploying data pipelines into an existing data pipeline service provided by Public clouds. AWS data pipeline service is a reference case.
Priority:	Must have
Affected Tools:	ORCHESTRATOR
Means of Verification:	Tests to verify that deployment to AWS data pipeline service is successful

ID:	R-T5.4-9
Type:	PORTABILITY
User story:	As an Operations Engineer, I want to deploy data pipelines to Private clouds

Requirement:	The data pipeline module of the Orchestrator should support deploying data pipelines expressed using TOSCA models into a private OpenStack cloud.
Extended Description:	The goal is to support deploying data pipelines in open source cloud platforms designed for setting up private clouds. OpenStack is chosen as the reference case. Open Source data pipeline platform (e.g. Apache NiFi) will be used as the runtime for the data
Priority:	Must have
Affected Tools:	ORCHESTRATOR
Means of Verification:	Tests to verify that deployment to private cloud platform is successful

ID:	R-T5.3-1
Type:	SECURITY
User story:	As an Operations Engineer, I want to define security and privacy policies of my application
Requirement:	The TOSCA blueprint needs to be able to support the definition of security and privacy policy of specific serverless/FaaS provider.
Extended Description:	The definition of security and privacy policy in TOSCA blueprint must be reflected after the deployment step is finished.
Priority:	Must have
Affected Tools:	DELIVERY_TOOLCHAIN
Means of Verification:	Test if security and privacy policy rules has been applied to deployed application by penetration testing methodologies.

ID:	R-T5.3-2
Type:	FUNCTIONAL_SUITABILITY
User story:	As an Operations Engineer, I want to define auto scaling policies of my application
Requirement:	The tool must be able to configure automatic scaling of the deployed components based on the auto scaling policies defined in the RADON models.
Extended Description:	The goal is to support policy-based (expressed in TOSCA blueprints) automatic scaling of deployed components, that are not fully managed by the cloud provider as long as the target platform supports automatic scaling. Examples of fully managed services are FaaS functions and managed database services.
Priority:	Must have
Affected Tools:	ORCHESTRATOR
Means of Verification:	Use simple load tests to verify that the deployed data pipeline tasks are scaled up and down according to the

	policy setting when autoscaling trigger thresholds are reached.
--	---

ID:	R-T5.3-3
Type:	PORTABILITY
User story:	As an Operations Engineer, I want to configure automatic scaling in Amazon EC2 cloud
Requirement:	The tool must be able to support configuring AWS EC2 auto scaling service based on the TOSCA auto scaling policy.
Extended Description:	The goal is to set up auto scaling for components that have been deployed in Amazon Ec2 as instances or containers. This will involve configuring CloudWatch alarms and AWS Auto scaling groups and rules based on TOSCA policies.
Priority:	Must have
Affected Tools:	ORCHESTRATOR
Means of Verification:	Use simple load tests to verify that the data pipeline tasks deployed to AWS EC2 are scaled up and down according to the policy setting when autoscaling trigger thresholds are reached.

ID:	R-T5.3-4
Type:	PORTABILITY
User story:	As an Operations Engineer, I want to configure automatic scaling in a Docker based environment
Requirement:	The tool should be able to support configuring automatic scaling of Docker services based on TOSCA auto scaling policies.
Extended Description:	The goal is to set up auto scaling for components that have been deployed as a service in a Docker swarm or Kubernetes. Depending on the target platform, it may be required to deploy a separate component for enacting the auto scaling decisions.
Priority:	Should have
Affected Tools:	ORCHESTRATOR
Means of Verification:	Use simple load tests to verify that the data pipeline tasks deployed to docker environment are scaled up and down according to the policy setting when autoscaling trigger thresholds are reached.

ID:	R-T5.1
Type:	COMPATIBILITY
User story:	As an Operations Engineer I want to use different FaaS/serverless providers at same time
Requirement:	The TOSCA blueprint must be able to enable Operations Engineers to define the usage of different FaaS/Serverless

	providers for different parts of their application.
Extended Description:	An Operations Engineer can decide to use different FaaS/serverless provider for different parts of his application in one RADON model. The delivery toolchain must support such deployment and monitoring of such application
Priority:	Must have
Affected Tools:	DELIVERY_TOOLCHAIN
Means of Verification:	Simple test if application is up and running should verify if deployment using mixed providers was successful.

ID:	R-T5.5-1
Type:	USABILITY
User story:	As an Operations Engineer I want my deploy stage to go through a CI tool so that my release is traceable
Requirement:	The xOpera command line interface needs to have a dry run mode to verify changes without asking for input in execution
Extended Description:	a) The --dry-run flag (or something similar) can give the user information about which changes will be activated. b) if the cli has input to verify. A workaround is to add a --auto-approve flag.
Priority:	Could have
Affected Tools:	DELIVERY_TOOLCHAIN, ORCHESTRATOR
Means of Verification:	Setting up a CI server

6. Use Case 1: Ambient Assisted Living UC

ID:	R-ENG-IoT-1
Type:	FUNCTIONAL_SUITABILITY
User story:	As a Software Designer I want to model in a graphical way the components of the SARA application
Requirement:	The RADON framework must enable the modelling of the components involved in the SARA application
Extended Description:	The different components of the SARA application (e.g Fall Detector, Location Service, Event Manager, Operator Web App etc.) and their deployment must be able to be described within the RADON model
Priority:	Must have
Affected Tools:	GMT
Means of Verification:	Demo based

ID:	R-ENG-IoT-2
Type:	SECURITY
User story:	As a Software Designer I want to express security constraints of the SARA application
Requirement:	The RADON framework must enable the modelling of

	security constraints in place between some components of the application
Extended Description:	The SARA application must address security constraints such as encryption of stored personal data, access to personal data by only authorized and authenticated users etc..
Priority:	Must have
Affected Tools:	CDL
Means of Verification:	Demo based

ID:	R-ENG-IoT-3
Type:	PERFORMANCE_EFFICIENCY
User story:	As a Software Designer I want to express performance constraints of the SARA application
Requirement:	The RADON framework must enable the modelling of performance constraints in place between some components of the application
Extended Description:	The SARA application must address performance constraints such as strong time constraints to guarantee real-time and low latency communications between the nodes of the system
Priority:	Must have
Affected Tools:	CDL
Means of Verification:	Demo based

ID:	R-ENG-IoT-4
Type:	RELIABILITY
User story:	As a Software Designer I want to express reliability constraints of the SARA application
Requirement:	The RADON framework must enable the modelling of reliability constraints in place between some components of the application
Extended Description:	The SARA application must guarantee a double connection with the Call Center (e.g. via a cellular connectivity and via Internet connectivity) when an alert message is sent
Priority:	Must have
Affected Tools:	CDL
Means of Verification:	Demo based

ID:	R-ENG-IoT-5
Type:	FUNCTIONAL_SUITABILITY
User story:	As a Software Developer, I want to accelerate the delivery of services in the nodes of the SARA application
Requirement:	The RADON framework must enable the modelling of the

	SARA event-driven environment to increase the capacity of the Cloe-IoT middleware through new serverless functions
Extended Description:	The RADON tools will be used to find the optimal deployment solution for the Cloe-IoT middleware to use serverless FaaS
Priority:	Must have
Affected Tools:	DECOMP_TOOL
Means of Verification:	Availability of new serverless functions triggered by IoT events

ID:	R-ENG-IoT-6
Type:	PERFORMANCE_EFFICIENCY
User story:	As a QoS Engineer I want to validate the achievement of the specified performance metrics
Requirement:	The RADON framework must enable the validation of the performance requirements of the SARA application
Extended Description:	In the SARA context the performance requirements are mainly related to response time of the components to alert messages
Priority:	Must have
Affected Tools:	TESTING_TOOL
Means of Verification:	Availability of performance test cases

ID:	R-ENG-IoT-7
Type:	FUNCTIONAL_SUITABILITY
User story:	As a Software Developer, I want to verify that the constraints of different SARA configurations are guaranteed
Requirement:	The RADON framework must support the verification of the different SARA configurations
Extended Description:	In the SARA application different configurations of the main nodes (ie the Body Area Network, the Robotic Rollator, the Robotic Assistant and the Smart Environment Gateway) are possible (eg Robotic Rollator is associated only to patients with walking problems) . The RADON tools will be used to check that the constraints of these configurations are guaranteed
Priority:	Must have
Affected Tools:	VERIFICATION_TOOL
Means of Verification:	Demo based

ID:	R-ENG-IoT-8
Type:	FUNCTIONAL_SUITABILITY
User story:	As a Software Developer, I want to identify defects on the topology of the SARA environment

Requirement:	The RADON framework must support the identification of defects on the topology of SARA environment in a graphical way
Extended Description:	The RADON tools will reduce the time to debug defects on the SARA infrastructure (e.g. connections between the nodes)
Priority:	Must have
Affected Tools:	DEFECT_PRED_TOOL
Means of Verification:	Demo based, time to detect defects reduced by 20% with regard to manual inspection

ID:	R-ENG-IoT-9
Type:	FUNCTIONAL_SUITABILITY
User story:	As a Software Developer, I want to collect and transfer into the cloud the SARA logs
Requirement:	The RADON framework must provide support of data pipelines for collecting and transfer the SARA log data into the cloud
Extended Description:	The log data collected from different nodes of the SARA application will be transfer into the cloud for data analytics tasks
Priority:	Must have
Affected Tools:	DATA_PIPELINE_PLUGINS
Means of Verification:	Availability of data pipelines for data logs

ID:	R-ENG-IoT-10
Type:	FUNCTIONAL_SUITABILITY
User story:	As a Software Developer, I want to deploy and orchestrate the software on the devices of the SARA application
Requirement:	The RADON framework must support the deployment and the orchestration of the SARA software
Extended Description:	The RADON tools will automate the deployment and orchestration of the software on the IoT devices of the SARA environment
Priority:	Must have
Affected Tools:	DELIVERY_TOOLCHAIN, ORCHESTRATOR
Means of Verification:	Deployment and orchestration of SARA software

7. Use Case 2: Managed DevOps UC

ID:	PRQ-SAM-INFRA-1
Type:	COMPATIBILITY
User story:	As a Software designer, I want to deploy Artifact Manager to two different cloud providers
Requirement:	The RADON framework must support deploying infrastructure

	to different cloud providers
Extended Description:	The two providers preferably being AWS and Azure.
Priority:	Must have
Affected Tools:	ORCHESTRATOR
Means of Verification:	Ability to deploy functional components in the required cloud providers without manual configurations

ID:	PRQ-SAM-INFRA-2
Type:	USABILITY
User story:	As a Software Designer, I want to model the necessary infrastructure components graphically
Requirement:	The RADON framework must support relevant components for graphical representation.
Extended Description:	Relevant to Praqma Cloud functions, events, storage, policies
Priority:	Must have
Affected Tools:	Winery/Che
Means of Verification:	Ability to recreate SAM with the Radon framework by using graphical models

ID:	PRQ-SAM-INFRA-3
Type:	PORTABILITY
User story:	As a Software designer, I want to model the necessary infrastructure components with code.
Requirement:	The RADON framework must support relevant components for representation of infrastructure as code.
Extended Description:	Relevant to Praqma Cloud functions, events, storage, policies
Priority:	Must have
Affected Tools:	Winery/Che
Means of Verification:	Ability to recreate SAM with Radon framework

ID:	PRQ-SAM-INFRA-4
Type:	COMPATIBILITY
User story:	As a Software Developer, I want to be able to create relations between components
Requirement:	The RADON framework should support the option of adding relations between components
Extended Description:	A relation can be a policy allowing a FaaS to insert content into a DB
Priority:	Must have
Affected Tools:	Winery/Che
Means of Verification:	Functions modelled with the RADON framework can be

annotated with constraints in the form of policies and these constraints are respected in the deployed FaaS.
--

ID:	PRQ-SAM-INFRA-5
Type:	RELIABILITY
User story:	As a Software Developer, I want to validate if there exists any relations between my cloud components
Requirement:	The RADON framework must provide information about any relations between components
Extended Description:	Describing the relations between S3 and a Function can for example be a trigger
Priority:	Must have
Affected Tools:	Winery/Che
Means of Verification:	Demo based

ID:	PRQ-SAM-INFRA-6
Type:	USABILITY
User story:	As a Software Developer, I want my modelled TOSCA Function to be executable as a blueprint
Requirement:	The RADON framework must be able to transform modeled components into a deployable blueprint
Extended Description:	xOpera should read models from Winery/Che
Priority:	Must have
Affected Tools:	xOpera
Means of Verification:	Demo based: xOpera can interpret a TOSCA modeled function from Winery/Che as an executable blueprint and then it can orchestrate and deploy that blueprint to the desired cloud provider.
Dependency:	Winery/Che

ID:	PRQ-SAM-INFRA-7
Type:	USABILITY
User story:	As a Software Developer, I want my executable blueprint to be viewed graphically
Requirement:	The RADON framework must be able to transform the deployable blueprint into a graphical/code based description of the model
Extended Description:	Winery should read blueprints from xOpera
Priority:	Could have
Affected Tools:	Winery/Che
Means of Verification:	Demo based: Given a blueprint generated by xOpera, Winery/Che will be able to view the blueprint graphically
Dependency:	ORCHESTRATOR

ID:	PRQ-SAM-INFRA-8
Type:	PORTABILITY

User story:	As a Software Developer, I want to be able to deploy the cloud infrastructure with CI/CD
Requirement:	The RADON framework should provide a CLI in order for a CI pipeline to deploy programmatically
Extended Description:	xOpera should have a CLI
Priority:	Must have
Affected Tools:	ORCHESTRATOR
Means of Verification:	Demo based: a TOSCA model in VCS (e.g. git) can be orchestrated from a CI server using xOpera.

ID:	PRQ-SAM-INFRA-9
Type:	PORTABILITY
User story:	As a Software Developer, I want to see the planned changes of a deploy before it takes a place
Requirement:	The RADON framework should provide a dry-run functionality to CLI
Extended Description:	The xOpera CLI should support a --dry-run flag
Priority:	Should have
Affected Tools:	ORCHESTRATOR
Means of Verification:	Demo based: When a TOSCA model is passed to xOpera with --dry-run, it will generate the planned changes of a deploy without actually applying any changes.

ID:	PRQ-SAM-INFRA-10
Type:	RELIABILITY
User story:	As a Software Developer, I want to verify the correctness of my models
Requirement:	The defect prediction tool should make me aware of any circular dependencies between my components
Extended Description:	RADON framework should flag any violations of the model constraints and any circular dependencies in the model
Priority:	Must have
Affected Tools:	Defect Pred tool
Means of Verification:	Demo based: RADON can flag circular dependencies in a model

ID:	PRQ-SAM-INFRA-11
Type:	COMPATIBILITY
User story:	As a Software Developer, I want to pass environment variables to functions
Requirement:	Being able to pass environment variables from CLI at runtime for the script to use
Extended Description:	Hardcoding secrets in scripts is a bad practice. Having env vars emits this risk
Priority:	Must have

Affected Tools:	ORCHESTRATOR
Means of Verification:	Running the CLI with "deploy --set Foo:bar"

ID:	PRQ-SAM-INFRA-12
Type:	FUNCTIONAL_SUITABILITY
User story:	As a Software Developer, I want to create an API gateway with specific paths related to subfunctions of my script
Requirement:	A declarative way of binding an http path to a sub function on a script
Extended Description:	Being able to use of script and bind "def updateDB(id)" to the http trigger "api/update/id"
Priority:	Must have
Affected Tools:	Template Library
Means of Verification:	Demo based

8. Use Case 3: Travel Technology UC

ID:	ATC-VIA-1
Type:	PERFORMANCE_EFFICIENCY
User story:	As a Software Designer, I want to be able to express performance efficiency constraints with respect to time and resource usage of my application
Requirement:	The RADON framework must be able to allow the specification of quality constraints on performance efficiency characteristics
Extended Description:	The VIAROTA quality constraints in terms of performance efficiency must be described in RADON models
Priority:	Must have
Affected Tools:	CDL
Means of Verification:	Demo based

ID:	ATC-VIA-2
Type:	RELIABILITY
User story:	A a Software Designer, I want to to be able to express reliability constraints for my application.
Requirement:	The RADON framework must be able to allow the specification of quality constraints on reliability characteristics
Extended Description:	The VIAROTA quality constraints in terms of reliability must be described in RADON models
Priority:	Must have
Affected Tools:	CDL
Means of Verification:	Demo based

ID:	ATC-VIA-3
Type:	FUNCTIONAL_SUITABILITY
User story:	As a Software Developer, I want to be able to graphically define the relations between the functions and the components of my application
Requirement:	The RADON framework must be able to allow model the app structure detailing components and functions
Extended Description:	RADON should graphically present the VIAROTA modelling
Priority:	Must have
Affected Tools:	GMT
Means of Verification:	Demo based

ID:	ATC-VIA-4
Type:	SECURITY
User story:	As a QoS Enginner, I want to be able to define and verify security and privacy concerns on data access and sharing on my app
Requirement:	The RADON framework must be able to allow the specification of quality constraints on security and privacy characteristics
Extended Description:	The VIAROTA security and privacy constraints must be described in RADON models
Priority:	Must have
Affected Tools:	CDL
Means of Verification:	Demo based

ID:	ATC-VIA-5
Type:	FUNCTIONAL_SUITABILITY
User story:	As a Software Developer, I want to be able to refactor my existing code to accommodate additional features for new development
Requirement:	The RADON framework must be able to allow the customisation of the app structure to minimise the dependencies to a monolithic approach
Extended Description:	RADON should graphically allow customising the VIAROTA modelling with new components, functions and relations
Priority:	Must have
Affected Tools:	GMT, DECOMP_TOOL
Means of Verification:	Demo based

ID:	ATC-VIA-6
Type:	COMPATIBILITY



Deliverable 2.1: Initial requirements and baselines

User story:	As a Software Developer, I want to be able to consume external APIs based on commonly used standards and practices
Requirement:	The RADON framework must be able to allow define different data structures and API formats in the app structure
Extended Description:	RADON should allow using common data formats in describing the relations to external functions
Priority:	Should have
Affected Tools:	FUNCTION_HUB
Means of Verification:	Demo based

ID:	ATC-VIA-7
Type:	PORTABILITY
User story:	As a Software Developer, I want to be able to transfer collected data in different cloud environments for multiple stakeholders
Requirement:	The RADON framework must be able to allow the specification of hybrid cloud hosting services
Extended Description:	RADON should support different data pipelines in cross cloud environments
Priority:	Must have
Affected Tools:	DATA_PIPELINE_PLUGINS
Means of Verification:	Demo based

ID:	ATC-VIA-8
Type:	PERFORMANCE_EFFICIENCY
User story:	As a QoS Engineer, I want to be able to validate quality metrics and KPIs prior to the release of my app
Requirement:	The RADON framework must be able to validate quality characteristics
Extended Description:	RADON should validate time and resource utilisation requirements
Priority:	Must have
Affected Tools:	TESTING_TOOL, VERIFICATION_TOOL
Means of Verification:	Demo based

ID:	ATC-VIA-9
Type:	RELIABILITY
User story:	As a QoS Engineer, I want to be able to identify any defects on my app structure prior to release
Requirement:	The RADON framework must be able to discover any defects and communicate them in a user friendly way
Extended Description:	RADON should verify the correctness of development with respect to any defects

Priority:	Must have
Affected Tools:	DEFECT_PRED_TOOL
Means of Verification:	Demo based, the tool detects more defects with regards to manual inspection

ID:	ATC-VIA-10
Type:	MAINTAINABILITY
User story:	As a Software Developer, I want to use a CD/CI approach for releasing new vesions of the new features of my app in the current operational version
Requirement:	The RADON framework must be able to guide my development teams in the CD/CI approach
Extended Description:	RADON should manage different teams in developing and deploying extensions of an existing app
Priority:	Must have
Affected Tools:	DELIVERY_TOOLCHAIN
Means of Verification:	Demo based

ID:	ATC-VIA-11
Type:	PORTABILITY
User story:	As a Software Developer, I want to deploy different instances of my application for different stakeholders
Requirement:	The RADON framework must be able to guide my development teams in the deployment of the app for different performace requirements
Extended Description:	RADON should automate the deployment of my app for different operational environments
Priority:	Must have
Affected Tools:	DELIVERY_TOOLCHAIN
Means of Verification:	Demo based