D2.2	
Version	1.0
Author	CNR
Dissemination	PUBLIC
Date	30-06-2018
Status	FINAL

D2.2 SotA revision document v1

Project acronym	ELASTEST
Project title	ElasTest: an elastic platform for testing complex distributed large software systems
Project duration	01-01-2017 to 31-12-2019
Project type	H2020-ICT-2016-1. Software Technologies
Project reference	731535
Project website	http://elastest.eu/
Work package	WP2 User-centered agile conception
WP leader	TUB
Deliverable nature	Report
Lead editor	Francesca Lonetti
Planned delivery date	30-06-2018
Actual delivery date	29-06-2018
Keywords	Open source software, cloud computing, software engineering, operating systems, computer languages, software design & development

Funded by the European Union

License

Contributors

Name	Affiliation
Andreas Afentakis	REL
Haytham Assem	IBM
Antonia Bertolino	CNR
Juan Caballero	IMDEA Soft
Antonello Calabrò	CNR
Anton Cervantes	ATOS
Malena Donato Cohen	ATOS
Guglielmo De Angelis	CNR
Felicita Di Giandomenico	CNR
Juan Angel Fuentes	NAEVATEC
Micael Gallego	URJC
Boni García	URJC
Nikolaos Stavros Gavalas	REL
Ilie-Daniel Gheorghe-Pop	FOKUS
Francisco Gortázar	URJC
Akis Gourgoulis	REL
Varun Gowtham	TUB
Piyush Harsh	ZHAW
Guiomar Tuñón Hita	NAEVATEC
Magda Kacmajor	IBM
Francesca Lonetti	CNR
Eda Marchetti	CNR
Enric Pages	ATOS
Michael Pauls	TUB
Frances Pavlidis	REL
Avinash Sudhodanan	IMDEA Soft

Version history

Version	Date	Author(s)	Description of changes
0.1	10/04/2017	Francesca Lonetti Antonia Bertolino	Initial ToC
0.2	15/06/2017	Francesca Lonetti Antonia Bertolino Guglielmo De Angelis Magda Kacmajor	First draft version of scientific SoTA
0.3	30/06/2017	ALL	First draft version of technical SoTA
0.4	15/07/2017	Malena Donato Cohen	First draft version of Market Analysis
0.5	09/01/2018	Francesca Lonetti Antonia Bertolino Guglielmo De Angelis	Revised ToC
0.6	10/04/2018	Francesca Lonetti Antonia Bertolino Guglielmo De Angelis Magda Kacmajor	Second draft version of scientific SoTA
0.7	30/04/2018	ALL	Second draft version of Technical SoTA
0.8	08/05/2018	Malena Donato Cohen	Second draft version of Market Analysis
0.9	20/05/2018	Francesca Lonetti Antonia Bertolino Guglielmo De Angelis	Revision of the document
0.10	24/05/2018	Francesca Lonetti	Restructuring of the document; revise text overall; add Introduction and conclusions sections
0.11	18/06/2018	Boni García	Internal review
0.12	19/06/2018	Francesca Lonetti Antonia Bertolino Guglielmo De Angelis	Revised document following internal review
0.13	26/06/2018	Boni García Francisco Gortázar	Final Review
1.0	28/06/2018	Francesca Lonetti	Final Version

Table of contents

List of figures

List of tables

Glossary of acronyms

Acronym	Description
CI (Continuous Integration)	It refers to a software development practice
CLI (Command Line Interface)	It refers to a console or text based representation in which the user types the commands to interact with an operating system or device
DOM (Document Object Model)	It refers to a standard for accessing documents
EBS (ElasTest Big-Data Service)	A test support service provided by ElasTest
ECE (ElasTest Cost Engine)	A test engine provided by ElasTest
EDM (ElasTest Data Manager)	A core component of ElasTest
EDS (ElasTest Device Emulator Service)	A test support service provided by ElasTest
EIM (ElasTest Instrumentation Manager)	A core component of ElasTest
EMP (ElasTest Platform Monitoring)	A core component of ElasTest
EMS (ElasTest Monitoring Service)	A test support service provided by ElasTest
EOE (ElasTest Orchestration Engine)	A test engine provided by ElasTest
EoL (End of Life)	It refers to the End-of-life of a product or service
EPM (ElasTest Platform Manager)	A core component of ElasTest
EQE (ElasTest Question & Answer Engine)	A test engine provided by ElasTest
ERE (ElasTest Recommendation Engine)	A test engine provided by ElasTest
ESM (ElasTest Service Manager)	A core component of ElasTest
ESS (ElasTest Security Service)	A test support service provided by ElasTest
ETM (ElasTest Tests Manager)	A core component of ElasTest
EUS (ElasTest User Impersonation Service)	A test support service provided by ElasTest

FOSS (Free Open Source Software)	This refers to software released under open source licenses
GBP (Group Based Policy)	It refers to OpenStack intent-driven declarative policy model that presents simplified application-oriented interfaces to the user
GUI (Graphical User Interface)	It refers to a type of user interface that allows users to interact with electronic devices through graphical icons and visual indicators
HDFS (Hadoop Distributed File System)	It referes to distributed, scalable, and portable file system written in Java for the Hadoop framework
IaaS (Infrastructure as a Service), PaaS (Platform as a Service) and SaaS (Software as a Service)	This refers to different models of exposing cloud capabilities and services to third parties
IAM (Identity Access Management)	It refers to the security and business discipline that provides IT managers with tools and technologies for controlling user access to critical information within an organization
IoT (Internet of Things)	It refers to a system of interrelated computing devices, and digital objects that are provided with unique identifiers and the ability to transfer data over a network without requiring human-to-human or human-to-computer interaction
IT (Information Technology)	It refers to the use of computers to store, retrieve, transmit, and manipulate data
NFV (Network functions virtualization)	It allows to abstract network functions, allowing them to be installed, controlled, and manipulated by software running on standardized compute nodes
OSS (Open Source Software)	It refers to a type of computer software whose source code is released under a license in which the copyright holder grants users the rights to study, change, and distribute the software to anyone and for any purpose
PoC (Proof of Concept)	It refers to a demonstration aiming to verify that certain concepts or theories

	have the potential for real-world application
QA (Quality Assurance)	It refers to a set of activities for monitoring the software engineering processes and methods used to ensure quality
QoS (Quality of Service)	It refers to non functional attributes of the system
REST (Representational State Transfer)	It refers to an architectural style for providing standards between computer systems on the web, making it easier for systems to communicate with each other
SDN (Software-defined network)	It refers to an architectural technology that enables application controlled programming and management of network resources in a dynamic and scalable manner
SiL (System in Large)	A SiL is a large distributed system exposing applications and services involving complex architectures on highly interconnected and heterogeneous environments
SNMP (Simple Network Management Protocol)	It refers to a popular protocol for network management that is used for configuring network devices on an Internet Protocol (IP) network
SOA (Service-Oriented Architecture)	It refers to an architectural style where applications are essentially a collection of services that communicate with each other
SOAP (Simple Object Access Protocol)	It refers to a messaging protocol specification for exchanging structured information in the implementation of web services in computer networks
SotA	State of the Art
SUT (Software under Test)	This refers to the software that a test is validating
T-Job (Testing Job)	We define a T-Job as a monolithic (i.e. single process) program devoted to validating some specific attribute of a system. Current Continuous Integration tools are designed for automating the

	execution of T-Jobs. T-Jobs may have different flavors such as unit tests, which validate a specific function of a SiS, or integration and system tests, which may validate properties on a SiL as a whole
TaaS (Testing as a Service)	It refers to an outsourcing model in which testing activities are performed by a service provider
TiL (Test in the Large)	A TiL refers to a set of tests that execute in coordination and that are suitable for validating complex functional and/or non-functional properties of a SiL on realistic operational conditions. We understand that a TiL can be created by orchestrating the execution of several T-Job
VNC (Virtual Network Computing)	It refers to a technology for remote desktop sharing and remote access on computer networks
VM (Virtual Machine)	It refers to an emulation of a computer system
VPN (Virtual Private Network)	It extends a private network across a public network, and enables users to send and receive data across shared or public networks as if their computing devices were directly connected to the private network

1 Executive summary

This deliverable presents the results of the scientific, technical and market scouting activity performed into Task 2.1 of WP2 in the first reporting period. It gives from one side a broad overview of the existing scientific and technical solutions on the topics covered by the project, and on the other side, an analysis of the market to understand market trends and opportunities for ElasTest. The purpose of Task 2.1 is to assess the potential strengths and weaknesses of ElasTest in comparison with existing techniques and tools, and to evaluate the extent to which the developed solutions will be suitable to be integrated and/or enhanced to achieve the objectives of ElasTest project, namely providing an open-source generic and extensible platform supporting end-to-end testing of large complex cloud systems.

The document includes three parts. The first part (Section 3) presents the scientific literature results. Specifically, since the main goal of ElasTest is to provide a cloud platform for testing of modern applications, we present the results of a systematic review of literature on the broad field of cloud testing in the last five years and a framework we propose for categorizing all relevant research in this context. Then, we focus on the main research topics of ElasTest and present the results of two surveys on cloud orchestration and on test recommendation. By the analysis of the scientific literature it is clear that ElasTest goes exactly towards the current research direction in cloud testing whereas test orchestration and test recommendation are challenging research topics of the project and very few approaches are targeting them in literature.

The second part (Section 4 and Section 5) of the document includes the results of a technical analysis of the state of the art, aiming to collect the most important tools that can be of interest for ElasTest according to the technological areas specified in the Description of the Action (DoA)[1]. For doing that analysis, we reviewed the grey literature classifying the tools according to 15 technological aspects and outlining the ElasTest progress in each of the identified aspects. Specifically, this analysis was performed into three main steps: a first overview of the state of the art was conducted in May 2007 and two updates in November 2017 and March 2018 respectively. The survey is a collective work from the consortium, in which each partner was responsible for the technical analysis of one or more identified aspects according to its expertise. The results of this technical analysis show the lack of a comprehensive cloud platform able to address both the design of complex tests and their execution for the validation of large software systems.

Finally, the last part of this document (Section 6) presents an overview of the most important IT market trends and shows that the demand of competitive solutions as well as the growing investments in cloud testing will have a positive impact on the development of ElasTest.

2 Introduction

2.1 Objectives

The main objectives of this deliverable deal with:

1. Presenting the state of the art about the scientific literature related to topics of interest of ElasTest. As stated in the DoA [1], ElasTest is a cloud platform for testing large systems, so we start our analysis of the scientific literature by presenting the main results of a systematic survey on cloud testing. Then, we focus on the two main research activities of the project that are: test orchestration and test recommendation and we present the first results of a survey on test orchestration and the results of a systematic survey of recommender systems applied to software testing.
2. Presenting the state of the art about the technological research addressed by the project. For this, we identified 15 technological aspects, covering the technological areas specified in the DoA [1] that are: Continuous Integration, Non-functional Testing, Security Testing, Monitoring, GUI Automation and Impersonation, Cloud Instrumentation, Data Ingestion, Dashboard Management, WebRTC Testing, Cross-browser Testing, Test Execution & Visualization, Mobile Testing, Test Management, Testing Framework and Virtualization. For each of these aspects, we present the technical state of the art, namely the most relevant tools, a comparison of these tools and the current technological progress of ElasTest with respect to the presented tools. A mapping of the ElasTest components to SotA aspects they are advancing is also presented. The goal of this technical analysis of SotA is to be comprehensive of all relevant tools belonging to the defined technical aspects of SotA. For this, this part of the document is very long. For quick reading, the reader can refer to Table 5, Table 7, Table 9, Table 10, Table 12, Table 14, Table 16, Table 18, Table 20, Table 22, Table 24, Table 27, Table 29, Table 31, Table 33, Table 34, Table 36, Table 38, Table 39, Table 41 presenting the comparison of tools in each SotA aspect and finally to Table 42 for the mapping of ElasTest components to SotA advancements in all the defined aspects.
3. Presenting the state of the art about market trends. The goal is to present a quantitative and qualitative assessment of the market and give an overview of market trends focusing on the IT areas in which ElasTest can later create impact.

2.2 Structure of the document

This document is structured as follows: in Section 3 we present an analysis of the scientific literature about cloud testing, test orchestration and recommender systems applied to testing. Then, in Section 4, we provide an analysis of the technical state of the art focusing on the technological aspects that are of interest in ElasTest. In Section 5, we present a mapping of ElasTest components to the advancements of ElasTest into these aspects. Section 6 shows how ElasTest is aligned with markets trends in the areas in which ElasTest can have an impact. Finally, Section 7 draws conclusions.

3 Analysis of scientific literature

3.1 Systematic survey on cloud testing

Cloud computing is an accepted paradigm for deploying applications and services. Businesses desire to achieve higher-level operational performance and flexibility while keeping the development and deployment costs as lower as possible. Cloud computing

has a high impact in software testing with the diffusion of a new testing paradigm known as Testing as a Service (TaaS). TaaS is a new service model, which provides static or dynamic on-demand testing services in the cloud and delivers them as a service to customers.

The main advantages of cloud-based testing deal with [2][3][45]:

- Reducing the testing costs and time. Cloud computing offers unlimited storage as well as virtualized resources and shared cloud infrastructures that can help to eliminate required computer resources and licensed software costs as well as to reduce the execution time of large test suites in a cost-effective manner.
- Performing on-demand test services to conduct large-scale performance and scalability online validation.
- Performing testing of dynamic and distributed applications leveraging multiple operating systems, multiple browser platforms as well as a large number of concurrent users.

Besides the many advantages in using cloud-based systems, there are yet many realistic problems of cloud service testing that need to be solved [2][4][45], such as on-demand test environment set-up, the heterogeneity and lack of standards in test tools as well as the need to assuring and assessing user privacy and security of cloud-based applications inside a third-party cloud infrastructure.

In this deliverable, we present the results of a systematic literature review on cloud testing, with the main objective to identify and categorize the relevant research papers on cloud-based testing. To this purpose, we developed a framework showed in Figure 1 used for classifying the selected primary studies into three (non-overlapping) categories that are: testing in the cloud, testing of the cloud and testing of the cloud in the cloud, and six areas that are: Test Perspective, Test Design, Test Execution, Test Objectives, Test Evaluation and Application Testing. Within each area we list several topics. More details of this framework are in Section 3.1.4. This systematic survey has been conducted following the guidelines for systematic reviews in software engineering research proposed by Kitchenham and coauthors [20]. Following these guidelines, our research methodology includes: defining research question (Section 3.1.1) and search query (Section 3.1.2), performing the search process (Section 3.1.3), classifying data (Section 3.1.4) and presenting results (Section 3.1.5). We refer to [5] for the whole results of the systematic survey on cloud testing.

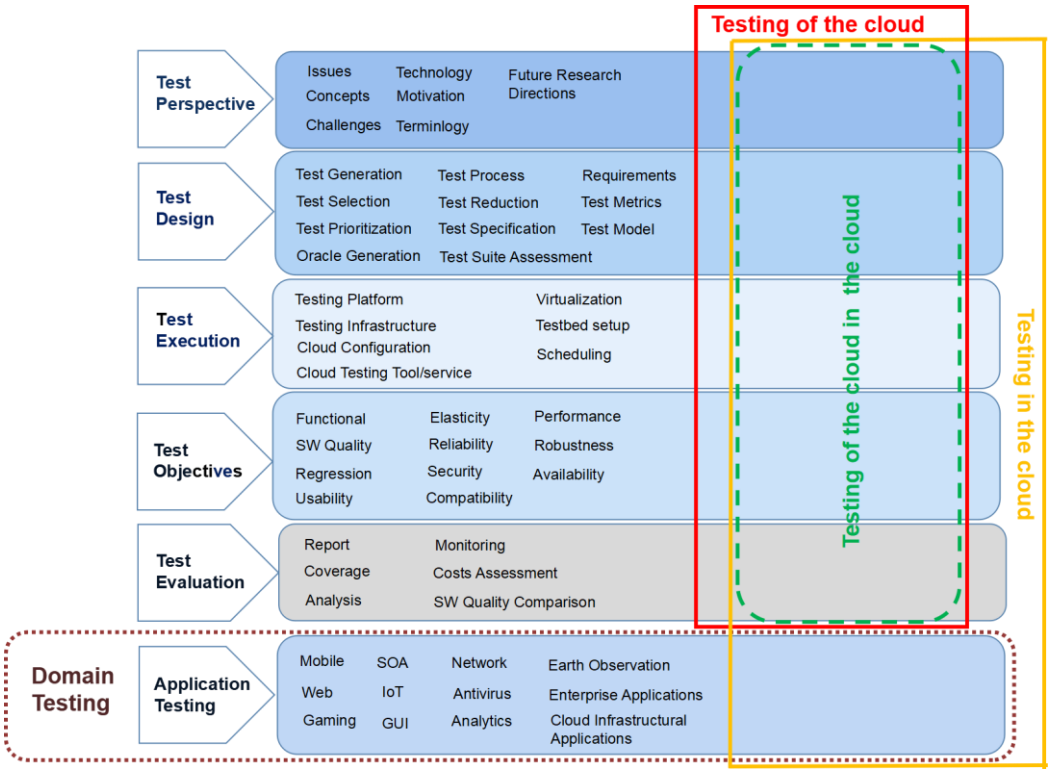

Figure 1. Cloud Testing Framework

3.1.1 Research questions

This review aims at summarizing the current state of the art in cloud testing research by proposing answers to the following questions:

1. What are the main objectives for cloud testing?
2. How are cloud resources exploited for software testing?
3. What are the test methods, techniques and tools mainly used in cloud-based testing?
4. How are testing results evaluated in cloud testing?
5. What are the research issues, challenges and future research directions of cloud testing?
6. Which are the main application domains for software testing in the cloud?

3.1.2 Research query

Based on the research questions, we defined the following search string. To be as comprehensive as possible, we defined a very general search string as shown below:

{test cloud} < OR > {cloud test} < OR > {testing over cloud} < OR > {cloud testing} < OR > {TaaS} < OR > {testing in cloud} < OR > {cloud-based test} < OR > {cloud-based testing} < OR > {testing in the cloud}

3.1.3 Search process

We performed an automated search for retrieving the more relevant primary studies dealing with cloud testing. Specifically, we searched by title, abstract and keywords in the following electronic sources selecting English papers from 2012 to 2017 and obtaining the following results.

- IEEE Xplore Digital Library– 134 results
- ACM Digital Library – 274 results
- Scopus – 247 results

We found a total of 655 peer-reviewed papers. We performed a first selection by reading title, abstract and keywords of the papers and selecting them according to some inclusion and exclusion criteria. Then, we performed a second selection of the included papers based on reading the whole paper and selecting them according to some quality assessment criteria. After this two-steps selection and a systematic snowballing procedure including both backward and forward snowballing, we found a final number of 147 primary studies.

3.1.4 Data extraction

Referring to Figure 1, we characterized papers according to three different categories of testing in a cloud environment:

- Testing in the cloud (yellow box in Figure 1) refers to software testing performed by leveraging scalable cloud technologies, solutions and computing resources. Papers belonging to this category present testing solutions for applications that can be developed in the context of different domains such as mobile or web environments and they are validated taking the advantage of large-scale test simulations and elastic computing resources on a cloud.
- Testing of the cloud (red box in Figure 1), i.e., validate the quality (functional and non-functional properties) of applications and infrastructures that are deployed in the cloud. Papers belonging to this category aim at checking the provided automatic cloud-based functional services, as well as at validating their performance, scalability, elasticity and security based on pre-defined SLAs.
- Testing of the cloud in the cloud (green box in Figure 1) refers to applications and infrastructures deployed in the cloud and tested in cloud environments. Papers belonging to this category fill the intersection area between Testing of the cloud and Testing in the cloud.

Moreover, we classified papers according to six different areas:

- Test Perspective. The papers belonging to this dimension present novel perspectives on cloud testing research.
- Test Design. The papers belonging to this dimension include all solutions targeting the design stage of testing activity such as test requirements, test model, test metric as well test cases generation and selection.

- Test Execution. The papers belonging to this dimension present artifact involved into the execution phase of the testing activity such as test platforms or tools or services.
- Test Objectives. The papers belonging to this dimension address the different purposes of cloud testing such as verifying that the systems comply with the functional specifications or that the system shows specific non-functional properties.
- Test Evaluation. The papers belonging to this dimension deal with the evaluation of the testing activity and test results.
- Application Testing. The papers belonging to this dimension present cloud-based testing solutions tailored specific application domains.

3.1.5 Summary of results

Figure 2 depicts the distribution of the primary studies per area and by years.

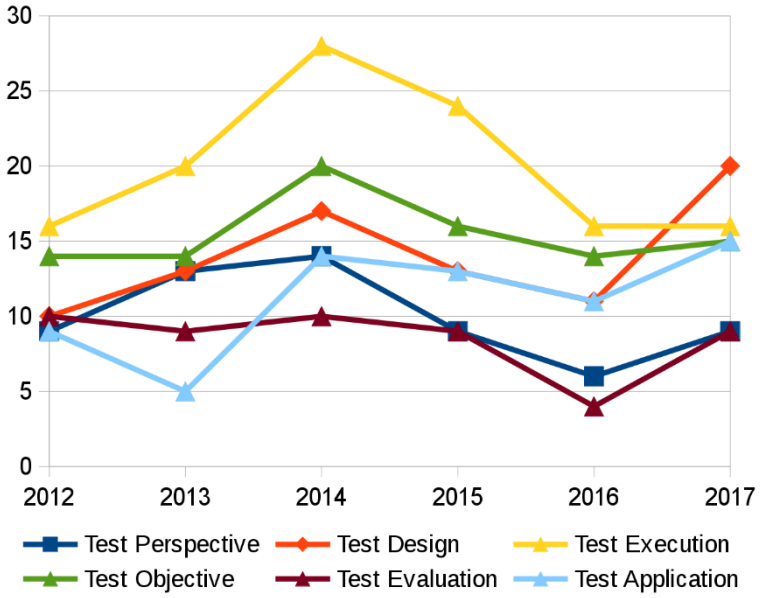

Figure 2. Trend of the Areas in the Primary Study by Year

As evidenced by this figure, the overall distribution of the papers in the considered period had a peak in 2014; on the other hand, there is a quite evident dominance of papers in the area of Test Execution. Then, since 2014, when the industry released more stable, and lightweight virtualization technologies, the number of academic publications starts decreasing. Nevertheless, during the last year of the observation period, it is possible to notice that the trend of the papers in some of the considered areas grows again.

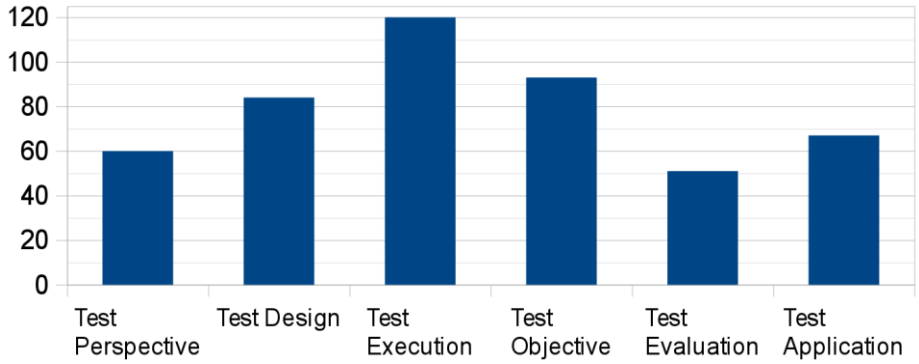

Figure 3. Distribution of the Papers by Area

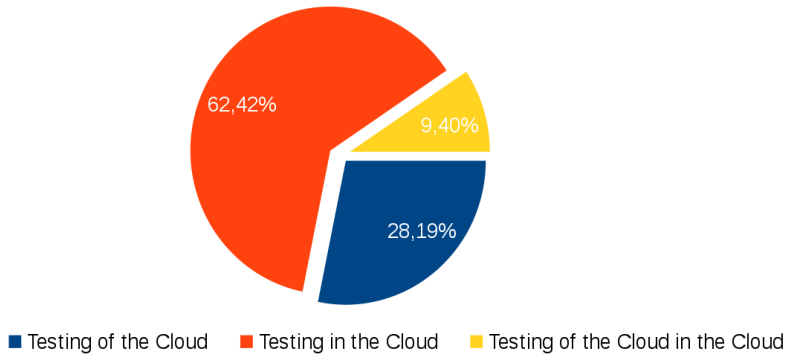

Figure 4. Partition of the Primary Study by Category

Figure 3 confirms such a trend by reporting the cumulative distribution of the Primary Study classified by Area.

Finally, Figure 4 depicts the distribution of the publications according to the category defined by the classification framework introduced in Section 3.1.4. Also, in this perspective, it results evident that most of the primary studies (i.e. 62.42%) addressed testing in the cloud solutions, while almost the 28% of the considered papers targeted to test a hosting cloud infrastructure. Finally, only a minor part of the primary studies (i.e. 9.4%) aimed to test cloud-based solutions when deployed over another underlying cloud platform.

Below we summarize the main results that answer to the research questions defined in Section 3.1.1.

Research question1: What are the main objectives for cloud testing?

The analysis of the target goals for cloud testing is given in Figure 5. We found that there is a considerable amount of primary studies covering functional testing, but most of the effort has been spent on approaches that validate performances attributes. This is due to the availability of many and relatively cheap resources in the cloud that makes performance testing easier. Finally, it is interesting to notice that also others non-functional objectives are covered by the analyzed primary studies: security (i.e. 10),

elasticity (i.e. 6), reliability (i.e. 6), robustness (i.e. 5), compatibility (i.e. 2), availability (i.e. 1), usability (i.e. 1), or in general software quality (i.e. 5).

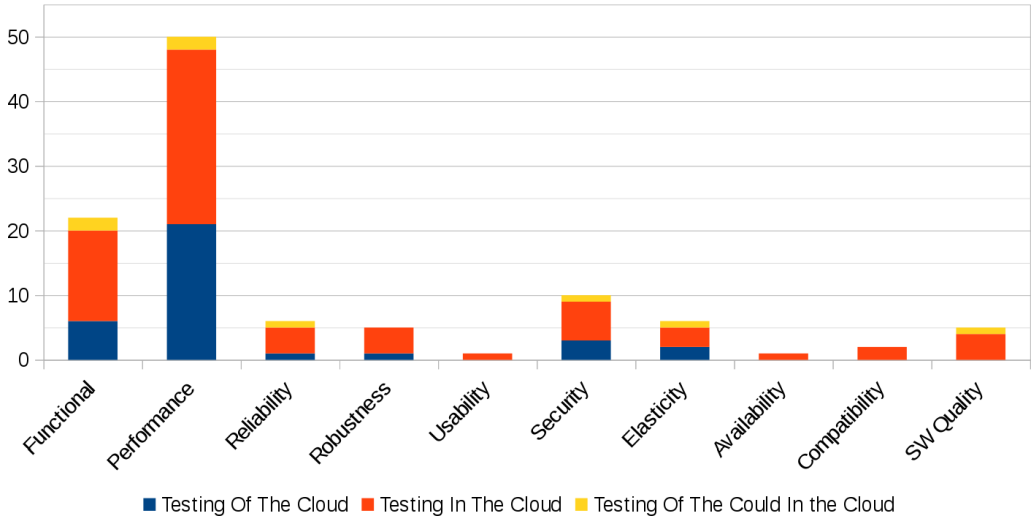

Figure 5. Breakdown of the Primary Study in Test Objective

Research Question2: How are cloud resources exploited for software testing?

Cloud infrastructures either public or private are used to provide TaaS according to a pay-per-use business policy [6]. Efficient resources allocation approaches and test scheduling solutions are introduced that maximize test resource utilization and improve load balance between resources [7]. Different strategies are proposed to i) partition the testing tasks; ii) allocate these testing tasks to different processors in the cloud platform for concurrent execution; iii) results collection from different processors. Some proposals focus on task decomposition methods and task scheduling algorithms in order to reduce the overall test time [8][9]. Dynamic resource adaptation strategies are defined to manage the cloud resources dynamically adding or removing virtual machines based on the workload of the cloud testing platform and the number of available devices. Finally, cloud resources are used in heavy testing techniques such as search based software testing including genetic algorithms. MapReduce is the most used model to easily process distributed data on multiple computers [10]. The goal is to exploit easy-to-use parallelization mechanism for exploring solution spaces larger than those considered by canonical sequential test techniques.

Research Question3: What are the test methods, techniques and tools mainly used in cloud-based testing?

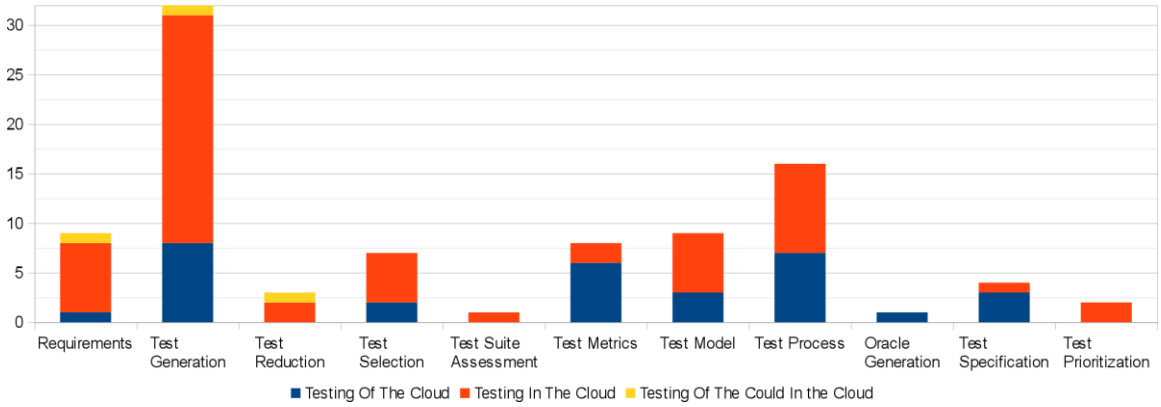

Figure 6. Breakdown of the Primary Study in Test Design

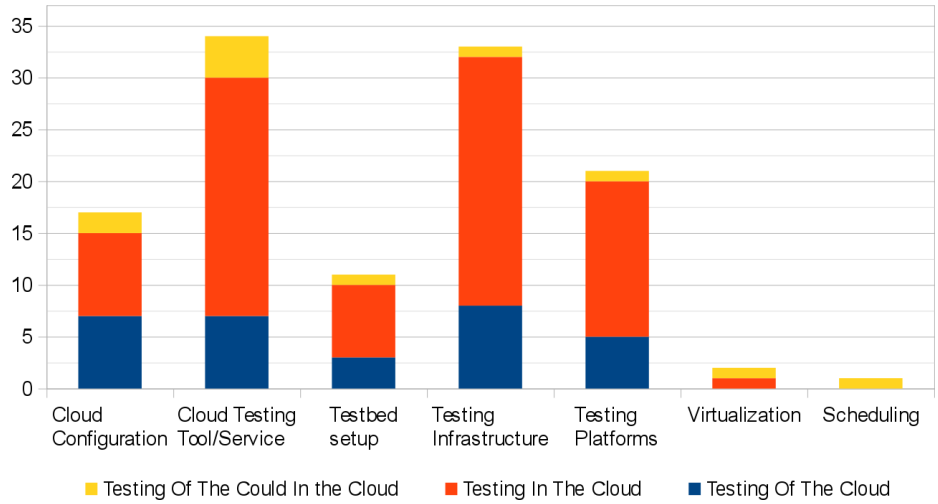

Figure 7. Breakdown of the Primary Study in Test Execution

Figure 6 highlights that among the 92 assigned items in the area of Test Design, the most addressed topics concerns Test Generation (i.e. 32), but also there is significant interest about the Test Process (i.e. 16). The items Test Model (i.e. 9), Test Metrics (i.e. 8), Requirements (i.e. 9), and Test Selection (i.e. 7) were sufficiently covered, while only marginal attention was deserved to Test Reduction (i.e. 3), Test Suite Assessment (i.e. 1), Oracle Generation (i.e. 1), Test Specification (i.e. 4), and Test Prioritization (i.e. 2). As a further consideration, a very limited number of primary studies in this area (i.e. 3) are specifically targeting to Testing of the Cloud in the Cloud. The breakdown of the primary studies in Figure 7 remarks the strong interest found with respect to Test Execution. Out of 121 expressed items in this area, several works present cloud testing tools or services (i.e. 34), testing infrastructures (i.e. 33) or platforms (i.e. 21), the configuration of cloud instances (i.e. 17). A minor part of the considered works discusses specific testbed setup in/for cloud environments (i.e. 11), others address virtualization (i.e. 2) or scheduling (i.e. 1) aspects.

Research Question4*:* How are testing results evaluated in cloud testing?

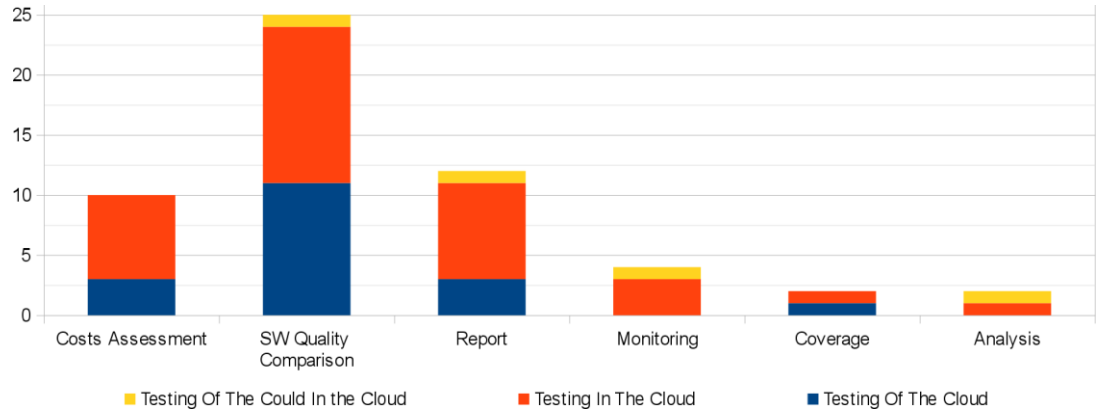

Figure 8. Breakdown of the Primary Study in Test Evaluation

As shown in Figure 8, the primary studies have been classified according to their capability of supporting the evaluation of the activities run in cloud testing. In this Area, most of the expressed tags (i.e. 25 over 55) concerns means for the comparison of quality attributes of the software-under test (SUT) in different conditions (e.g. configuration, deployment, load, etc.). Other items in Test Evaluation such as monitoring (i.e. 4), coverage (i.e. 2), and analysis (i.e. 2) received a marginal consideration; these results enforce the idea that cloud testing is perceived as the modern promise for quantitative analysis of the SUT, but also that a considerable attention is given to those methods able to feed the outcomes of the technological experimentation into proper methodological frameworks.

Research Question5*:* What are the research issues, challenges and future research directions of cloud testing?

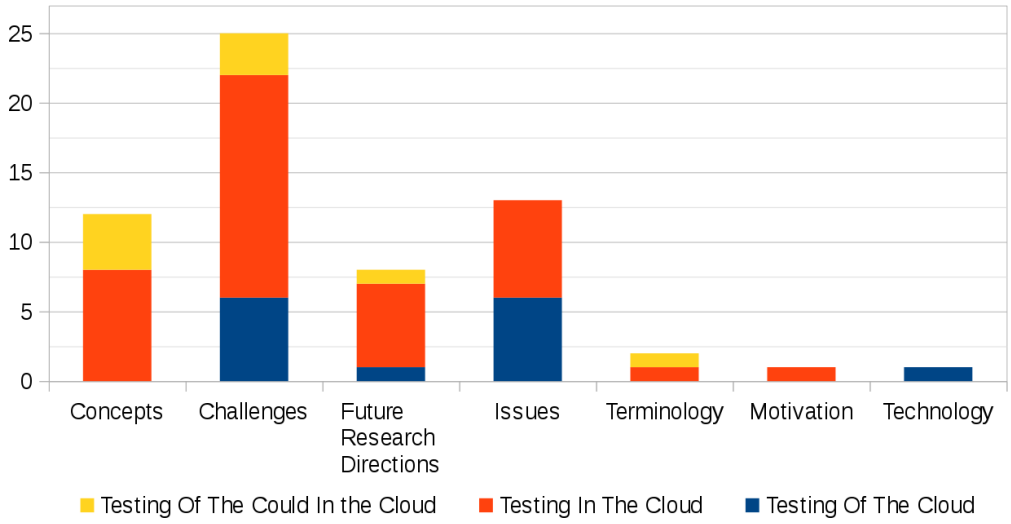

Figure 9. Breakdown of the Primary Study in Test Perspective

Figure 9 reveals that most of the interest for these primary studies was in describing the challenges subsumed by the cloud testing approaches (i.e. 25), but also several primary

studies mainly gave a presentation about the concepts of the paradigm (i.e. 12) and its potential issues (i.e. 13). A minor number of primary studies directly addressed future research direction for cloud testing (i.e. 8), while only a few are focused on aspects such as terminology (i.e. 2), technologies (i.e. 1), or motivations (i.e. 1).

Research Question6*:* Which are the main application domains for software testing in the cloud?

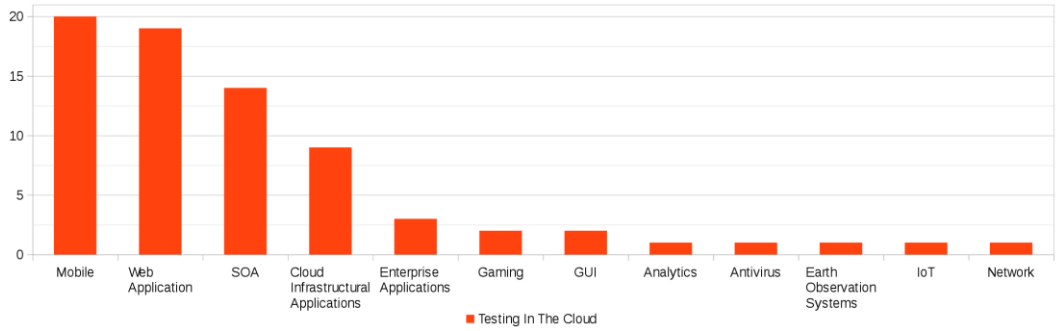

Figure 10. Breakdown of the Primary Study in Test Application

Figure 10 reports the classification of the target domain of primary studies aimed to testing a software/application in the cloud. Within this category, the collected data confirm that researchers leverage on cloud-based testing mostly for validating web application (i.e. 19) or software specifically targeted to mobile devices (i.e. 20); nevertheless, also a relevant number of works referred to the cloud as a mean for testing solution abiding by SOA paradigm (i.e. 14). In addition, the review also found a discrete number of papers (i.e. 9) addressing cloud infrastructural applications (e.g. multi-tenancy and data distribution infrastructures), and several attempts in exploiting the perspectives of the cloud in order to narrow testing activities on specific application domains such as: antivirus, earth observation, enterprise applications, gaming, GUI, high workload data analytics, IoT, networks emulation.

3.1.6 Conclusions

By this systematic review, it was evident that in the recent years a lot of attention has been devoted to the cloud testing. Indeed, the facilities and resources that cloud environment makes available to the different users open the path for innovative and more effective solutions and rise new challenges for the testing activity.

As evidenced by this survey, the research activity has been mostly focused on test design, test generation and test execution, since cloud offers new opportunity to develop and maintain costly test infrastructures and to leverage on demand scalable resources. In this context, ElasTest will advance the state of the art by providing a comprehensive cloud platform able to address both the design of complex tests and their execution for the validation of large software systems, particularly in what refers to non-functional properties such as reliability, security and scalability. Moreover,

ElasTest will improve the effectiveness of the current testing process by introducing two new test concepts that are ElasTest Orchestration and ElasTest Recommendation. We present a survey on test orchestration in Section 3.2 and a systematic survey on recommender systems applied to software testing in Section 3.3.

3.2 Test Orchestration

According to the ElasTest DoA [1], test orchestration is one of the three main research directions of the project. The test orchestration notion as defined in the context of the ElasTest project is completely new in the literature. We started a survey on test orchestration and we provide some preliminary results in this section. We plan to make a more complete systematic survey along the remaining part of the project. In the following sections, we describe the most important approaches found in this survey (Section 3.2.1), a comparison among them (Section 3.2.2) and the progress into ElasTest project (Section 3.2.3).

3.2.1 Preliminary results

We identified the following main approaches that are close to ElastTest orchestration concept. Table 1 provides a brief description of these papers.

Source	Short Description
[11]	The paper presents an extension of Buildbot architecture (consisting of a single buildmaster and one or more buildslaves, connected in a star topology) aiming at providing tests orchestration during continuous development. The main goal of the approach is to select platform and configuration information of executed tests, select tests to execute and their order according to build information and finally making the analysis of tests reports so that developers can quickly identify errors if any. Tests are triggered through UI, eclipse plug-in or java application and are organized in different queues for each type of test. Main advantages of the proposal are automated test configuration and execution as well as parallel tests execution. It allows to run both functional and non-functional tests and to integrate different testing tools such as Junit and Jenkins.
[12]	The paper proposes an orchestration framework to facilitate the automated testing of Ajax-based web applications. The framework provides the following main capabilities: building test objects,

	test cases generation, test injection, monitoring and reporting. A tool in PHP code has been developed implementing the main functionalities of the architecture such as crawling, test cases generation and execution.
[13]	This paper provides a testing framework using a compositional approach to generate and execute test cases. In this framework, the requirements are expressed as logical formulas derived by a set of abstract predicates. Elementary abstract test cases are derived from these requirements and instantiated using the SUT (Software Under Test) knowledge to obtain concrete test cases. The correct orchestration of elementary tests execution and combination of their results is given by the test generator deducing the validity of the overall logical formula. Specifically, from a logical formula a test generation function automatically produces a tester consisting of a set of communicating test controllers, one for each operator appearing in the formula. Tests are then executed to obtain the final verdict. A prototype tool for Java environments, implementing the proposed approach is also presented.
[14]	The paper provides the design of a framework for orchestrating automated testing and deployment. The aim is to help to automate the code analysis, test selection, test scheduling, execution and results analysis by using a deployment pipeline. This framework uses a build-deploy-test-report workflow on the target platform to deploy and test software application when it runs a build. The main goal is to create an end to end pipeline that can orchestrate the appropriate builds, environment provisioning, testing and test results analysis and deployment steps.
[15]	This work focuses on hierarchical testing in order to target failing functions avoiding exhaustive test execution. Tests are dynamically composed using test graph notation. At each stage of testing process, after tests are run, an analysis of results is done to decide which tests will be performed in the next cycle, using policies based

	on software testing maturity level, for making unfolding of a suitable sub-graph of tests. The goal is to avoid execution of tests not contributing further information to already executed tests, with consequent time saving that increases with software system size.

Table 1. Test Orchestration - First Results

3.2.2 Results Comparison

Testing orchestration is a very challenging research topic, and a very few approaches are targeting it in literature. The approaches we found in our survey can be classified according to the following main dimensions which are of interest in ElasTest as shown in Table 2.

Source	SUT specification	Test cases orchestration notation	Functional Test	Non- functional tests	Oracle specification	Test report	Testing tools	Continuous integration	Tool
[11]	-	-	X	X	-	X	JUnit, Jenkins	X	-
[12]	-	-	X	X	Integrates predefined oracle	X	Crawljax, soapUI, Sahi,Junit, PHPUnit	-	X
[13]	X	X	X	-	Derives a final verdict	-	-	-	-
[14]	-	-	-	-	-	X	JUnit, Nunit	X	-
[15]	-	X	-	-	-	-	-	-	X

Table 2. Comparison of test orchestration results

SUT specification: It refers to a formal or informal specification of behavioral requirements of the system under test. This specification is used for deriving test cases. Many approaches do not cope with such SUT specification whereas in [13] these requirements are expressed by logical formulas build upon abstract predicates describing operations performed on the system under test.
Test cases orchestration notation: It represents the notation used for defining tests orchestration rules. It consists of a logical formula in [13] whereas it is a graph-based formalism in [15].
Functional vs Non-functional *tests*: It refers to the kind of tests that are supported by the proposed solution. Some approaches deal with both functional and non-functional testing whereas for other works it is not specified.
Oracle specification: With this concept we mean the capability of the testing approach to derive the expected output of the orchestrated test cases. In [12] the orchestration integrates a pre-configured oracle as part of the framework repository whereas in [13] a final verdict is obtained by the execution of the test cases on the implementation. No work is able to derive a test oracle specification from the combination of expected and obtained outputs of the executed test cases.
Test report*:* Almost all the approaches provide test reports allowing analysis of test results. These reports can be visualized in the browser after test execution or stored in the asset repository.
Testing tools: It refers to the used testing tools in the presented frameworks. The most used tool for generating test cases is Junit. SoapUI is used for generating test cases in soap request format whereas Jenkins is used to automate the testing procedure.
Continuous integration: It means the capability of the testing framework to support continuous integration and deployment. Some orchestration frameworks leverage a build-deploy-test workflow to deploy software product to a target system environment and run tests on them as part of the build process. Then, for each commit, the whole application will build and a set of automated tests will run against it. This allows identifying and fixing problems as soon as they occur.
Tool: It refers to the support provided by the different architectural solutions in terms of automated facilities for testing orchestration. Few approaches such as [12] and [15] provide full automated facilities for test generation and execution or composition. These tools are devoted to specific kind of systems as Ajax-based web applications or they are still limited in the implemented testing functionalities.

3.2.3 Progress within ElasTest

All proposals about testing orchestration focus on orchestrating the different phases of the testing process including test operation, test injection, monitoring and reporting, sometimes according to response-time, bandwidth-usage, throughput and adaptability. The goal is to integrate testing in the software development process in a pipeline approach with the aim of continuous integration and deployment. ElasTest will allow orchestration of testing units for deriving tests of large systems. ElasTest will go beyond the state of the art by providing a general test orchestration topology notation and orchestration rules applied to large systems that are now missing in the literature.

At the current status of the project (M18) we propose two different types of test orchestration, which we refer to as verdict-driven and data-driven. The former notion of orchestration does not assume any constraints or model availability on the TJobs whereas in the second approach TJobs can be interconnected using their test data (input) and their outcome (output). Both approaches are supported by the ElasTest Orchestration Engine (EOE). Specifically, leveraging the DSL notation of Jenkins pipelines, both for verdict and data-driven approaches we have implemented a Jenkins shared library which exposes a simple API to orchestrate jobs.
The ElasTest orchestration engine under development in WP4 will include and combine advances in topics that are part of the orchestration strategy such as test prioritization, selection, parallelization, tests dependency. For a review of these related topics, we refer to D4.1 Test Orchestration basic toolbox v1 [38].
We aim to create a complete theory around test orchestration concept as well as to provide further validation of the proposed test orchestration approaches.
Finally, ElasTest will target the problem of automated partial oracle derivation of large systems by inferring test oracle specification from the composition of expected and obtained outputs of the executed testing units.

3.3 Systematic survey on Recommender systems applied to software testing

A recommender system is a software application that supports users facing the challenge of vast information spaces and choice overload. This kind of challenge is certainly pervasive in software engineering, and a variety of Recommender Systems for Software Engineering (RSSEs) have been developed to address specific problems arising in this domain. The progress in the development of RSSEs over the last 10 years has been captured in several reviews ([16], [17], [18],[19]); however, the picture emerging from these publications reveals little about recommender systems applied to software testing phase. Pakdeetrakulwong et al. [18], who review RSSEs from software development lifecycle perspective, identify only two recommenders designed to support testers, while results of a comprehensive systematic survey presented in [19] include only one such solution. Two other reviews do not mention testing domain at all.

In this report, we present the results of a systematic literature review on recommender systems applied to software testing domain. The structure of the survey follows the recommendations for performing systematic literature reviews in software engineering proposed in [20]. Thus, our first step was to define the goal of the survey, followed by specifying the research questions. These are presented in section 3.3.1. Next, the research query (Section 3.3.2) was formulated and run against the relevant databases, as described in section 3.3.3. The next stage of the review was data extraction, as per categories listed in section 3.3.4, followed by data synthesis, whose summary results are presented in section 3.3.5.

3.3.1 Research questions

The goal of the survey has been formulated according to GQM template [21] as Table 3:

Name	Description
Object of study	Recommendation systems applied to software testing domain
Purpose	Characterize
Focus	Technique/algorithm used for generating recommendation
Stakeholder	Software test engineer
Context factors	Type of testing task, required data

Table 3.GQM template

The focus of the survey is on the methods used for generating recommendations, and the two crucial context factors are the type of data being manipulated, and the type of the task being supported. Thus, the research questions have been formulated as follows:

1. What types of problems/tasks are addressed by recommender systems applied to testing domain?
2. What approaches/algorithms are used for generating recommendations?
3. What sources of information (types of data) are exploited?
4. What is the input of the user? (Query/trigger for generating a recommendation)
5. What is the output produced by the recommender?

3.3.2 Research Query

As can be noticed in the previous studies [16][18][19], queries oriented to retrieving publications on RSSE (Recommender Systems for Software Engineering) in general, tended to elicit very few papers on recommenders applied to software testing. We formulated our query to target specifically the testing domain as in the following:

("recommendation systems" OR "recommendation system" OR "recommender systems" OR "recommender system") AND ("software testing" OR "software tester" OR "software testers" OR "software tests" OR "test case" OR "test cases")

3.3.3 Search process

In order to obtain sufficient amount of data, the query was run against 5 digital libraries, and the search was performed on full documents with the following results:

- IEEE Xplore Digital Library1 – 18 results
- ACM Digital Library2 – 17 results
- ScienceDirect3 – 39 results
- Scopus4 – 56 results
- Engineering Village5 – 55 results

After excluding: (i) papers not describing recommender systems applied to software testing domain, judged based on the abstract reading; (ii) secondary studies; and (iii) duplicates, the total amount of 18 documents was left for the analysis.

Given the focus of the research (technique/algorithm used for generating recommendations for testers), we have not limited the output to papers describing fully implemented solutions.

3.3.4 Data extraction

The following data categories have been extracted:

1. Purpose / issue addressed
2. Approach/algorithm for generating recommendations
3. Source of information (type of data as the basis for deriving recommendations)
4. User input/query to elicit a recommendation
5. Output produced

3.3.5 Summary of the results

Research question 1: What types of problems/tasks are addressed by recommender systems applied to testing domain?

There is no single most useful task/problem categorization. During data synthesis, we decided to consider the following aspects:

1. Testing phase (Figure 11). More than a half of the papers (10 in total) present solutions supporting either unit testing or regression testing. Only one publication refers to functional testing, and one – to performance testing. The remaining solutions do not focus on a single test phase.
2. User's task to be supported by the recommenders (Figure 12). The recommendations are most frequently applicable for (i) the task of test case selection and prioritization when creating a test plan to be executed in a future test

[1] http://ieeexplore.ieee.org

[2] https://dl.acm.org/

[3] http://www.sciencedirect.com

[4] https://www.scopus.com/

[5] https://www.engineeringvillage.com

campaign (7 papers), and (ii) the task of writing new test code (6 papers). Two other tasks are classified as gathering bug information (including creating bug reports), and testers allocation (3 and 2 papers, respectively).

3. System's task. (Figure 13). This category refers to automated tasks performed by the system in order to generate recommendations. These are typically sub-tasks of more general users' tasks described in point 2. After synthesizing the extracted information on underlying tasks performed by the recommender systems, we identified two most common types of tasks: i) Identifying error-prone software artifacts (source code modules or system settings), or test cases testing such artifacts (7 papers). Error-proneness is usually assessed based on frequencies of code changes, history of previous failures and code properties. An interesting approach described in [29] proposes exploiting user reviews for identifying critical test cases. ii) Finding test code to reuse (5 papers). The term "test code" refers not only to full test cases, but also to single test steps, test inputs, test oracles or other code snippets.

   The types of system's tasks can be almost exactly mapped onto user's tasks identified in point 2, which means that tester's tasks tend to be supported in a rather standard way.
4. High level purpose/benefit (Figure 14). The main driver for creating recommenders applied to software testing domain can be broadly defined as resource optimization (9 papers). This includes recommendations about task prioritization and task assignment. Test code reusability and gathering bug information were described in 4 and 3 papers, respectively.

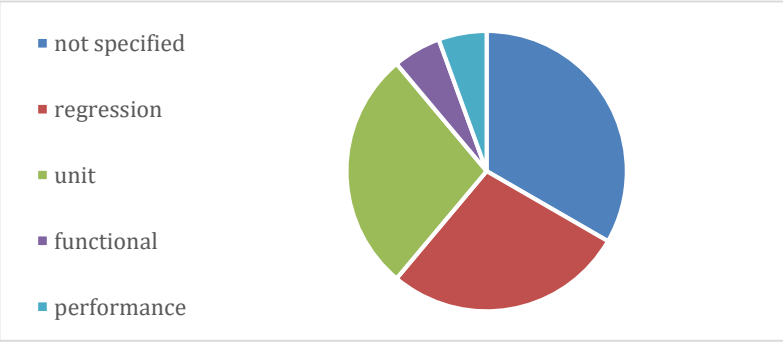

Figure 11. Testing phases the recommenders focus on

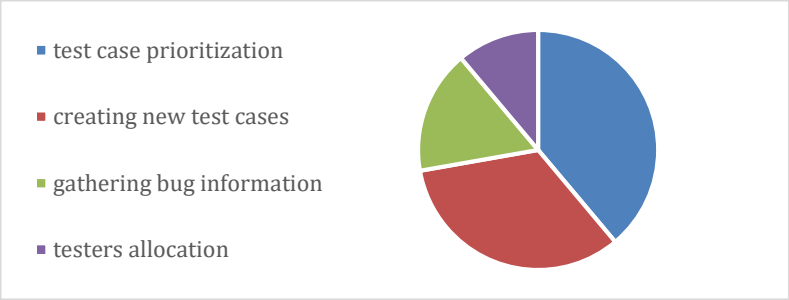

Figure 12. User's tasks supported by the recommenders

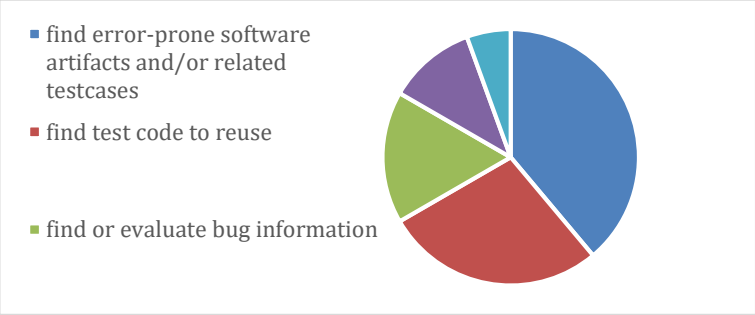

Figure 13. Tasks performed by recommender systems in order to generate recommendations

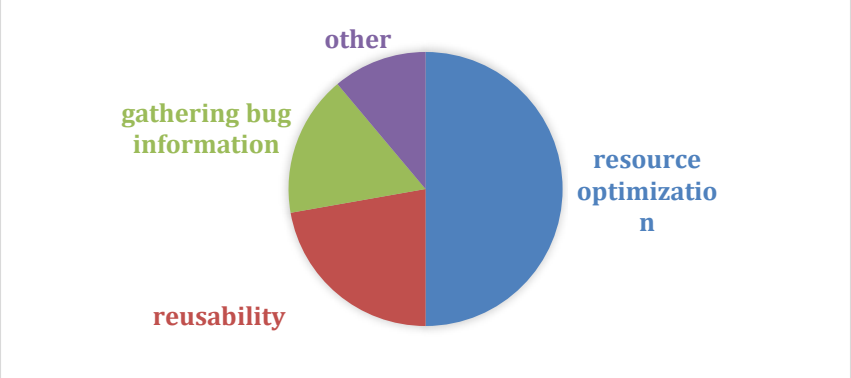

Figure 14. High level purpose/benefit of using recommenders

Research question 2: What approaches/algorithms are used for generating recommendations?

Main approaches identified during data synthesis are presented in Figure 15, together with the number of publications mentioning each approach. Among NLP (Natural Language Processing) tasks, we identified syntactic parsing, NER (Named Entity Recognition) and information extraction using bag-of-words representations. Machine learning approaches are limited to K-means clustering, hierarchical clustering, Random Forests, linear regression and SVM. Source or test code mining approaches include clone detection, test-driven search and association rule mining.

In Figure 14 we omitted methods mentioned by just one of the reviewed publications, such as ontology building, multivariate classification (Mahalanobis-Tagushi system), greedy optimization algorithm or FixCache algorithm.

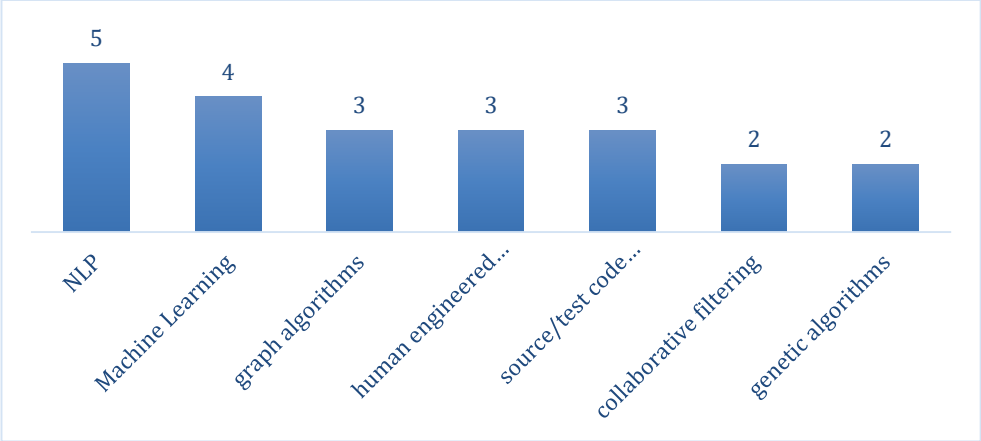

Figure 15. Techniques used for generating recommendations. Top labels refer to the number of publications mentioning given method

***Research question 3**:* What sources of information (types of data) are exploited?

A typical recommender system presented in the reviewed publications relies on several sources of information. As shown in Figure 16, the required knowledge is predominantly derived by parsing or mining the production source code (as opposed to the test code). Second most popular basis for generating recommendations is the history of changes applied to the source code. Roughly one-third of solutions exploits unstructured text (user stories, bug reports, requirements, manual test scripts) or semi-structured text (execution traces, Gherkin scripts), while one-fourth of recommenders takes advantage of prior test executions records. 7 publications in total describe solutions that pull data from test cases repository: 4 systems extract the information by parsing or mining test code, while 3 others utilize test cases' properties (e.g. test phase, version number or creation time).

The chart in Figure 16 shows sources of information that were indicated in at least three studies. Less common types of data (not included in the chart) were class diagrams, properties of system-under-test (namely, OS configuration), survey results and user evaluations.

In addition, six solutions required meta-information, i.e. some sort of mapping or traceability information between different types of data, such as the mapping between source classes or requirements and the corresponding test cases.

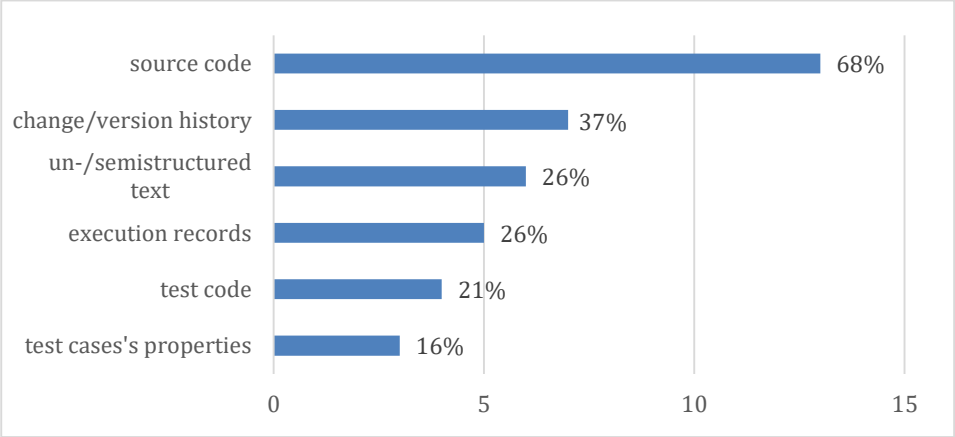

Figure 16. Data exploited by test recommender systems

Information utilized by test recommenders can be also classified with regard to whether it is internal to the software development project or sourced from the outside world. We found that only three solutions reach for the external data: [22] and [24] utilize cross-project data retrieved from open web repositories, and [29] proposes exploiting product evaluations extracted from online reviews. All the remaining recommenders are entirely dependent on data specific to the software project under test.

Research question 4: What is the input of the user? (Query/trigger for generating a recommendation)

For the majority of recommenders (10 papers) the required input is the code under test – a class under test, a snippet of class to be tested, or the code for the whole release, along with the change history. Six systems accept unstructured or semi-structured text as the input. In [28], the input is a bug report to be improved, and [23] and [26] require a concrete test suite information and available testers profiles. [27] introduces the concept of fat query, which encompasses the whole context of a bug, such as past bug reports, logs, change history, core dumps and information about people who can be consulted.

Research question 5: What is the output produced by the recommender?

Not surprisingly, the outputs produced by the recommenders are aligned with system's tasks described in results for Research Question 1 (point 3) – that is, recommenders designed for test cases prioritization task return ranked list of test cases (or items that can be directly mapped to existing test cases, such as error-prone classes or configurations). One exception is [25], which returns a numeric score of importance and error-proneness for a single class under test. In a similar vein, systems whose task is finding reusable code return test code, and those designed to find best match between testers and test cases – output testers allocation.

3.3.6 Conclusions

Despite running the research query against multiple digital databases, and applying relatively lenient selection criteria, the number of publications relevant for the survey was low, indicating that in the testing domain, recommender systems are under-represented.

Furthermore, recommendations described in the elicited publications represent two broad classes: code recommendations and test cases prioritization recommendations. Within each class, the scope of the offered solutions is limited.

Regarding the first class, recommendations are limited to code reuse, leaving room for new solutions involving code generation. Within the second class, tests are recommended and prioritized according to the error-proneness criterion rather than requirements importance. In the context of the well-known distinction between "doing things right and doing the right thing", these systems can be perceived as focusing on the first aspect, "doing things right" – in this case, improving the quality of code. However, "doing the right thing" – in this case, meeting customer requirements – is at least equally important aspect, which seems to be rarely addressed by the existing solutions.

The main contribution from ElasTest to this area is building personalized recommender system for facilitating the tester to design test cases more efficiently by recommending the testing developer with the specific T-Job combinations to be included into a TiL. As evidenced in this survey, this type of recommender systems is completely missing in the current SotA. To achieve this novel direction, there is a need to progress beyond the SotA in the combination of advanced supervised approaches, based on structured input/output along with unsupervised and semi-supervised learning. The recommender system will overcome the problem of modeling rare unseen events of SiL by exploiting recent advances in deep learning [30][31][32]. One major challenge in supervised learning is the availability of supervised data, which is expected to be expensive and difficult to gather in this situation as well due to typically requiring expert knowledge for manually labeling the data produced from testing the SiL. In order to tackle this challenge, we will also apply: (i) semi-supervised learning that combines larger unsupervised data (e.g., SiL specs, business requirements) with small set of supervised data (T-Job relevancy), and (ii) domain adaptation, which allows for learning functions that can perform well in the different domains. We refer to D4.2 Test Recommendation Engines v1 [44] for advancements of ElasTest in this research direction.

4 Technical Analysis of SotA

In this section, we focus on technical analysis of SotA, namely the most important tools that can be of interest for ElasTest. They have been classified according to the following main aspects covering the technological areas in which ElasTest will advance the state of the art, as stated also in the DoA [1]:

- Continuous Integration
- Non-functional Testing
- Security Testing

- Monitoring
- GUI Automation and Impersonation
- Cloud Instrumentation
- Data Ingestion
- Dashboard Management
- WebRTC Testing
- Cross-browser Testing
- Test Execution & Visualization
- Mobile Testing
- Test Management
- Testing Framework
- Virtualization

The analysis includes a different number of tools for each aspect and has been performed along all the first reporting period. Specifically, a first overview of the state of the art was conducted in May 2007 and two updates in November 2017 and March 2018 respectively. We describe the most important tools for each aspect in the following sections.

4.1 Continuous Integration

In this section, we provide the state of art about continuous integration. Specifically, we divide the tools into four main categories: Continuous integration Server; Continuous integration - Identity Access Management; Continuous integration - Artifact distribution; Continuous integration – Docker Image Distribution, as in the following.

4.1.1 Continuous integration Server - Baseline and comparative analysis

In this section, we present in Table 4 the available tools for continuous integration server, then we describe and compare the most relevant ones.

Name	URL	Brief description	Licence
Apache Continuum	http://continuum.apache.org/	It is an enterprise-ready CI server with features such as automated builds, release management, role-based security, and integration with popular build tools and source control management systems.	OSS (Apache 2.0 license)
Apache Gump	http://gump.apache.org/	It is a CI system aimed to build and test Java projects. It supports Apache Ant, Apache Maven (1.x to 3.x) and other build tools.	OSS (Apache 2.0 license)

AppVeyor	https://www.appveyor.com/	It is a distributed CI service used to build and test projects hosted at GitHub on a Microsoft Windows virtual machine.	Proprietary
Atlassian Bamboo	https://www.atlassian.com/software/bamboo	It is a CI server used to automate the release management for a software application, creating a continuous delivery pipeline.	Proprietary but provides free license for OSS
AWS CodeBuild	https://aws.amazon.com/es/codebuild/	AWS CodeBuild is a fully managed build service that compiles source code, runs tests, and produces software packages that are ready to deploy.	Proprietary
Buddy	https://buddy.works/	It is a web-based and self-hosted CI/CD server used to build, test and deploy web sites and applications with code from GitHub, Bitbucket and GitLab.	Proprietary
Buildbot	http://buildbot.net/	It is an open-source framework (CI/CD/release management) for automating software build, test, and release processes.	OSS (GPL)
BuildMaster	http://inedo.com/buildmaster	It is an Application Release Automation tool used to create simple or complex release pipelines managed from one central dashboard.	Proprietary
Cabie	http://cabie.tigris.org/	It is a Multi-platform client/server-based application providing both command line and web-based access to real time build monitoring and execution information.	OSS (GPLv2)
CruiseControl	http://cruisecontrol.sourceforge.net/	CruiseControl is both a CI tool and an extensible framework for creating a custom continuous build process.	OSS (BSD-style)

Drone	https://github.com/drone/drone	Drone is a continuous integration and continuous delivery platform built on container (Docker) technology, written in Go. Every build is executed inside an ephemeral Docker container, giving developers complete control over their build environment with guaranteed isolation.	OSS (Apache 2.0 license)
GitLab CI	https://about.gitlab.com/gitlab-ci/	It is an integrated continuous integration and continuous delivery server to test/build/deploy GitLab repositories.	OSS (MIT Expat license)
GoCD	https://www.gocd.io/	It is a continuous delivery tool that supports automating the entire build-test-release process from code check-in to deployment.	OSS (Apache 2.0 license)
Hudson	http://hudson-ci.org/	Hudson is a continuous integration (CI) tool written in Java, which runs in a servlet container, such as Apache Tomcat or the GlassFish application server. It supports Software Configuration Management (SCM) tools including CVS (Concurrent Version System), Subversion, Git and Clearcase and can execute Apache Ant and Apache Maven based projects, as well as arbitrary shell scripts and Windows batch commands.	OSS
Jenkins	https://jenkins.io/	Jenkins is a self-contained, open source automation server which can be used to automate all sorts of tasks such as building, testing, and deploying software.	OSS (MIT)

OpenMake Software	https://www.openmakesoftware.com/	Set of tools: Openmake Meister for Build Automation; Openmake Release Engineer for Multi-Platform Application Release Automation (ARA).	Proprietary
Team Foundation Server	https://www.visualstudio.com/tfs/	Microsoft tool that provides source code management (Team Foundation Version Control or Git), reporting, requirements management, project management, automated builds, lab management, testing and release management capabilities.	Proprietary
TeamCity	http://www.jetbrains.com/teamcity/	Java-based build management and CI server from JetBrains.	Proprietary
Travis CI	https://travis-ci.org/	Hosted, distributed CI service used to build and test software projects hosted at GitHub.	OSS (MIT)
Vexor	http://vexor.io/	Distributed cloud web-service for building and testing CI tool.	Proprietary
Visual Studio Team Services	https://www.visualstudio.com/vso/	It provides continuous integration and delivery capabilities to an Azure web app, VM, container, Xamarin Test Cloud, HockeyApp, etc.	Proprietary

Table 4. Continuous integration server tools

The most relevant tools about continuous integration server according to the opinions registered on G2Crowd[6] and Slant[7] are: AppVeyor, Atlassian Bamboo, CruiseControl, Drone, GoCD, Jenkins, Team Foundation Server, TeamCity, Travis CI.

In the following sections, we present a brief description of each of them.

[6] Open Source Initiative, licenses https://opensource.org/licenses/alphabetical

[7] Slant.co https://www.slant.co/topics/799/~best-continuous-integration-tools

4.1.1.1 *AppVeyor*

AppVeyor[8] claims to be the most popular continuous delivery service for Windows. It's a hosted, distributed continuous integration service used to build and test projects hosted at GitHub on a Microsoft Windows virtual machine. It focused on applications developed with .Net.
AppVeyor is configured using a Web UI, or by adding a file named appveyor.yml, which is a YAML format text file, to the root directory of the code repository.
AppVeyor has a free trial and different pricing plans whereas discount is provided on all plans for students, educational organizations and open-source projects looking for more calculation power or concurrent jobs.

4.1.1.2 *Atlassian Bamboo*

Bamboo[9] is Atlassian's continuous delivery and release management tool. It offers continuous delivery, tying automated builds, tests, and releases together in a single, integrated workflow. Bamboo gives developers, testers, build engineers, and system administrators a common space to work and share information – while still keeping sensitive operations like production deploys locked down. It integrates with many CI tools and works better with other Atlassian tools as JIRA and BitBucket.
Bamboo integrates with GitHub and supports multiple projects formats and coding languages.
Atlassian offers a suite of tools for the development of Open Source projects free of charge while these requirements are met: i) the project is licensed under a license approved by the Open Source Initiative; ii) the project source code is available for download; the open source project has a publicly accessible website; Atlassian's software is accessible to the public.

4.1.1.3 *CruiseControl*

CruiseControl[10] is both a continuous integration open source tool and an extensible framework for creating a custom continuous build process. It includes dozens of plugins for a variety of source controls, build technologies, and notifications schemes including email and instant messaging. A web interface provides details of the current and previous builds but it is old fashioned and not intuitive enough. The standard CruiseControl distribution is augmented through a rich selection of 3rd Party Tools.

4.1.1.4 *Drone*

Drone[11] is a Continuous Integration platform built on container technology. Every build is executed inside an ephemeral Docker container, giving developers complete control over their build environment with guaranteed isolation.

---

[8] AppVeyor https://www.appveyor.com/

[9] Atlassian Bamboo https://www.atlassian.com/software/bamboo

[10] CruiseControl http://cruisecontrol.sourceforge.net/

[11] Drone https://github.com/drone/drone

The open source edition is actively maintained and was re-built from scratch to use the Docker engine. The closed sourced edition previously powering drone.io is no longer being maintained.
The actual open source drone directive is to help teams ship code like GitHub. Drone is easy to install, setup and maintain and offers a powerful container-based plugin system. Drone aspires to eventually offer an industry-wide replacement for Jenkins.
At the moment of this writing, the open source project status is Beta.

4.1.1.5 *GoCD*

GoCD [12] is an open source tool (heir to CruiseControl) which is used in software development to achieve continuous delivery (CD) of software. It supports the automation of the entire build-test-release process from code check-in to deployment. It helps to keep producing valuable software in short cycles and ensure that the software can be reliably released at any time.
GoCD is easy to install and run, but quite complex in order to set jobs, task and pipelines dependencies. Once configured, it has a friendly interface and enables to visualize results and jobs in a pretty way.
It has many plugins that enable integration with other CI tools, but some CI common tools are not yet integrated.

4.1.1.6 *Jenkins*

Jenkins[13] is an open source automation server with an unparalleled plugin ecosystem to support practically every tool as part of the delivery pipeline. It allows to automate continuous integration processes (and other kind of processes). Processes are defined as jobs that can be launched by timers, polls or manually.
Jenkins is easy to install, and fairly easy to use for configuring jobs, authentications, permissions. Jenkins has a big pool of available plugins that allow Jenkins to be personalized in many ways and integrated with the most used CI tools.
Jenkins is a good established CI server, and widespread within developers and testers communities.

4.1.1.7 *Team Foundation Server*

Team Foundation Server (TFS) [14] is an enterprise-grade server for teams to share code, track work, and ship software for any language, all in a single package. TFS works better as a part of the Microsoft CI tools, but multiple tools (proprietary and open source) can be integrated with TFS.
Team Foundation Server is available in two different forms: on-premises and online.
TFS is quite easy to install and configure while working within the Microsoft Suite and Microsoft Cloud platform (Azure).

[12] GoCD https://www.gocd.io/

[13] Jenkins https://jenkins.io/

[14] Team Foundation Server https://www.visualstudio.com/tfs/

4.1.1.8 *TeamCity*

TeamCity[15] is a continuous integration and deployment server from JetBrains. It is known to be quite harsh to setup, but once it is done, building projects using Ant or MSBuild is incredibly easy, whereas for many other languages the initial setup of the build configuration can be a bit daunting. As many environments are supported such as Java or Python, TeamCity shines while used in .Net environments.
TeamCity provides a wide range of additional plugins to provide integration with most used CI tools.

4.1.1.9 *Travis CI*

Travis CI [16]is a hosted, distributed continuous integration service used to build and test software projects hosted at GitHub.
Travis CI can be configured to run the tests on a range of different machines, with different software installed (such as older versions of a programming language implementation to test for compatibility) and supports building software in numerous languages as well as integration with many external tools.
Configuration of jobs within Travis CI is quite specific and needs each project to contain a specific file to set the CI server job configuration. Travis CI is 100% free for Open Source projects, with source code hosted in GitHub. Over 300k open source projects and 235k users are testing on Travis CI.

Table 5 shows a comparison of CI server tools according to some common dimensions such as simplicity of usage, acceptance by the developer community, expertise of partners, integration with other CI tools, costs.

	AppVeyor	Bamboo	Cruise Control	Drone	GoCD	Jenkins	Microsoft TFS	TeamCity	Travis CI
Simplicity of usage	Good	Very Good	Bad	Good	Bad	Good	Good	Medium	Good
Hosted	Yes	No	No	No	No	No	Yes/No[17]	Yes/No[18]	Yes
Supported types of projects	.Net	multiple	multiple	multiple	multiple	multiple	multiple	multiple	multiple
Integration with other CI tools	Poor	Good	Average	Good	Average	Very Good	Good	Good	Good
Acceptance by the	Niche	High	Low	Medium	Low	High	Medium-High	Low	Free

[15] TeamCity http://www.jetbrains.com/teamcity/

[16] Travis CI https://travis-ci.org/

[17] TSF has both kind of licenses.

[18] Hosted for eligible OSS projects.

developer community									
Expertise of partners	Low	Low	Low	Low	Low	High	Low	Low	Average
Costs	High	Free License[19] +Hosting	Open Source + hosting	Open Source + hosting	Open Source + hosting	Open Source + hosting	High	Average (License + Hosting)	Free

Table 5. Comparison of CI server tools

4.1.2 Continuous integration - Identity Access Management - Baseline and comparative analysis

Identity management, also known as identity and access management (IAM) is, in computer security, the security and business discipline that "enables the right individuals to access the right resources at the right times and for the right reasons". It addresses the need to ensure appropriate access to resources across increasingly heterogeneous technology environments and to meet increasingly rigorous compliance requirements. It is desirable for all the tools under the CI process to share a unique IAM, so just one account for user would be necessary to access all the tools with the appropriate permissions. Table 6 shows the available tools for Continuous integration - Identity Access Management. Most relevant tools are: Google Sign-In, GitHub API and OpenID server. They are described in the following.

Name	URL	Brief description	License
AWS Identity and Access Management (IAM)	https://aws.amazon.com/es/iam/	It allows for securely control access to AWS services and resources. It only works within AWS but can federate users from an external identity provider (OpenID and SAML).	Proprietary
OAuth (GitHub)	https://developer.github.com/v3/oauth/	OAuth2 is a protocol that lets external applications request authorization to private details from an user's GitHub account without revealing their password.	Free SW
Google Sign-In	https://developers.google.com/identity/	Google Sign-In is a secure authentication system that reduces the burden of login for users, by enabling them to sign in with their Google account—the same account	Free SW

[19] Requirements must be met.

		they already use with Gmail, Play, Google+, and other Google services.	
OpenID	http://simpleid.koinic.net/	OpenID server can be deployed to register to all (or most) the tools we should use for CI.	OSS

Table 6. Continuous integration - Identity Access Management tools

4.1.2.1 *Google Sign-In*

Google Sign-In[20] enables developers to use its OAuth2 service to provide its applications with a trusted login. This simplifies development and design as it relays the user validation to Google. This benefits users as the user can use just one account for login in multiple applications. Google provides users the possibility to check at any moment which applications are logged in with his/her Google account, and to revoke any authorization.
Many of the possible CI tools we will be using within ElasTest project can be configured to use Google Sign-In. As not all partners share Google domain and some of the partners have company policies against the use of some of Google Apps, Google Sign-In wouldn't be a suitable choice to manage the authentication on ElasTest CI tools.
Google Sign-In supports 2-phase authentication but it does not provide a method to applications which use its API to force 2-phase authentication to the account.

4.1.2.2 *GitHub API*

GitHub[21] also provides an OAuth2 service for logging to external applications. It is based on application registration into the GitHub. The application could be registered under a GitHub organization so all the members of the Organization are allowed to use GitHub while logging in on that application, without necessity of configuring all the parameters on their own GitHub account.
GitHub provides users the possibility to check at any moment which applications are logged in with his/her google account and revoke any authorization.
Many of the possible CI tools that will be used within ElasTest project can be configured to use GitHub as login provider.
As GitHub is the source repository selected to enable collaboration for all public components, it makes sense to use the GitHub login and the already existing ElasTest GitHub organization to log-in in other CI tools.
GitHub login supports 2-phase authentication, and it can be forced for users in an organization.

[20] https://developers.google.com/identity/protocols/OpenIDConnect

[21] GitHub Login (OAuth2) https://developer.github.com/v3/oauth/

4.1.2.3 *OpenID server*

Another possibility to centralize most of the CI tools we will be using in the ElasTest project can be an own managed OpenID[22] server where all the collaborators could register. The main reasons for hosting an OpenID server would be: i) we would want to use the ElasTest URL site without setting up complex delegation rules; ii) we would want finer control over partners' and collaborators' identity configuration than that provided for in other web services; iii) other OpenID providers or log-in providers don't fulfil security and/or privacy policy of some partners. Hosting our own OpenID server would lead to extra costs in order to host and maintain the OpenID server. Also, with our own OpenID server activating 2-phase authentication won't be trivial and will result in another extra cost.

Table 7 shows a comparison of Identity Access Management tools according to some common dimensions such as simplicity of usage and setup, access and permission management, authentication type, integration with other CI tools and costs.

	Google Log-In	GitHub OAuth	OpenID server
Simplicity of usage	High	High	High
Simplicity of setup	High	High	Low
Access and Permission management	Very Good	Very Good	Custom
Two-phase Authentication	Yes	Yes	Yes[23]
Integration with other CI tools	High	High	High
Costs	Free	Free	variable on the Cloud provider
Other Characteristics	Some clauses can be conflictive with privacy and security policies of partners	GitHub has been already selected as code repository	

Table 7. Comparison of CI - Identity Access Management tools

[22] http://simpleid.koinic.net/

[23] It is possible to set up a two-phase authentication but it is not trivial, and it will result in extra costs.

While Google Login is possibly one of the most integrated authentication methods on a wide range of CI tools, we believe there would be some kind of incompatibility with some partner's privacy and security policies. So, the best option found is GitHub OAuth. This option seems to fit better on the current state of the project, as GitHub has already been selected as a code repository, so each of the partners should have a GitHub user. Also, an ElasTest organization and teams have been created to manage permissions.
The OpenID server should be the last choice mainly because of its cost and the difficulty of setting two-phase authentication. This option would be chosen only in case any partner has strong restrictions on the use of GitHub as login provider.

4.1.3 Continuous integration - Artifact distribution - Baseline and comparative analysis

Table 8 shows the available tools for Continuous integration - Artifact distribution. Among them, the most relevant ones are: Maven Central and Nexus Repository.

Name	URL	Brief description	License
Gitlab	https://gitlab.com/	GitLab is a web-based Git repository manager with wiki and issue tracking features, using an open source license, developed by GitLab Inc.	Freeware
Maven Central	http://repo.maven.apache.org/maven2/	Central repository for free distribution artifacts	Freeware
Nexus Repository OSS	https://www.sonatype.com/download-oss-sonatype	The world's first and only universal repository solution that is free to use.	Freeware

Table 8. CI- Artifact distribution tools

They are software tools designed to optimize the download and storage of binary files used and produced in software development. We will focus on Java-based artifacts, and more exactly on Maven/Gradle driven projects. As ElasTest will be composed by private and public components, public and private artifact repositories will be studied, and two different tools may be selected in order to satisfy both types of distribution needed within the project. These tools are described below whereas Table 9 presents a comparison between them. The comparison is focused on these aspects of the different tools: simplicity of usage, access and permission management, public/private access, acceptance by the developer community, previous knowledge and experience of partners, and integration with other CI tools and costs.

4.1.3.1 *Maven Central*

Maven Central[24] is the main public artifact repository for Java-based artifacts. All maven driven projects use this repository for public artifact repository. Only big companies or large OSS projects maintain public repositories separately from Maven Central, for example Atlassian or JBoss have their own maven repository but even those have their public releases accessible from Maven Central.

4.1.3.2 *Nexus Repository OSS*

Nexus Repository[25] is the most used open source artifact repository. It supports many binary formats and it is fully integrated with maven driven projects. It allows developers to proxy, collect, and manage project dependencies. It makes it easy to distribute artifacts. Internally, builds (within the CI server) can be configured to publish artifacts to Nexus and they then become available to other developers.
The cost of maintaining a private artifact repository is related to the Cloud Provider selected and the space necessities for it, the license for the tool is free under the Creative Commons License[26].

	Maven Central	Nexus Repository OSS
Simplicity of usage	High	High
Access and Permission management	--	Good
Acceptance by the developer community	High	High
Previous knowledge and experience of partners	High	High
Integration with other CI tools	High	High
Costs	Free	Hosting and storage

Table 9. Comparison of CI - Artifact distribution tools

[24] Maven Central http://central.sonatype.org/

[25] Nexus Repository https://www.sonatype.com/nexus-repository-oss

[26] https://creativecommons.org/licenses/by-nc-nd/3.0/us/

4.1.4 Continuous integration – Docker Image Distribution - Baseline and comparative analysis

Usually, the places to store and distribute Docker images are called Docker Registry. They are content delivery and storage systems for named Docker images. It can be thought of as a collection of repositories keyed by name. There are multiple providers that offer Docker Registries, public and private. Also, it is possible to create and host a custom-made Docker Registry. The most relevant tools for public/private docker image distribution are: DockerHub, Quay, Amazon EC2 Container Registry.
As ElasTest will be composed by private and public components, public and private Docker Registries will be studied, and two different tools may be selected in order to satisfy both types of distribution needed within the project. Below a description of the existing tools, whereas Table 10 shows a comparison among them. As for CI - Artifact distribution tools, the comparison is focused on the following aspects of the different tools: simplicity of usage, access and permission management, public/private access, acceptance by the developer community, previous knowledge and experience of partners, integration with other CI tools, Costs.

4.1.4.1 *DockerHub*

DockerHub[27] is public Docker Registry with possibility to host public and private Images. For public Images it is completely free of charge.
This Registry is widespread within the community of developers, and many well-known OSS projects have there their official images. Some examples are: Nginx, Ubuntu, Apache Tomcat. DockerHub is really easy to use as it is the predefined Registry while working with docker. This will provide a good way to share the public images with the community. DockerHub supports both teams and organizations. Owners of an organization have administration rights for all the repositories of the organization. Then created teams can be granted specific permissions (write/admin) on each of the repositories. The access to public repositories within DockerHub is “read” for all the DockerHub users. The integration possibilities with this registry are multiple as it isn't necessary for any tool to specifically manage the integration with DockerHub as the integration with Docker already provides the necessary capacities for using DockerHub as registry.

4.1.4.2 *Quay*

Quay[28] is a well-known public Docker Registry with the possibility to host public and private Images. Working with Quay within Docker is easy enough, although isn’t as straightforward as working with other technologies (e.g. DockerHub). Also, Quay supports organizations and teams, it provides many ways of integration, as Quay works not only as a registry but also as an image builder and image security tester, these are extra functionalities that could be of use on ElasTest.

[27] DockerHub https://hub.docker.com/

[28] Quay https://quay.io

4.1.4.3 *Amazon EC2 Container Registry*

Amazon EC2 Container Registry[29] (ECR) is a fully-managed Docker container registry that makes it easy for developers to store, manage, and deploy Docker container images. It can be viewed as an intermediate solution between a custom-made Registry and registry services as DockerHub and Quay.
ECR transfers images over HTTPS and automatically encrypts the images. The integration of ECR with other CI tools is good, as Amazon has launched multiple plugins to simplify this integration.

	DockerHub	Quay	Amazon ECR
Simplicity of usage	High	Medium	High
Access and Permission management	Very Good	Very Good	Custom
Acceptance by the developer community	High	Medium	--
Previous knowledge and experience of partners	Average	Low	Average
Integration with other CI tools	High	High	High
Costs	Low	Medium	Medium
Other Characteristics	Some clauses on Terms of Service can be conflictive with privacy and security policies of partners	Extra Features Possibility of discontinuation	While many of the CI tools are ran on AWS EC2 within the same region costs drop significantly

Table 10. CI docker image distribution comparison

4.1.5 Progress within ElasTest

As at first sight DockerHub could fulfil most of the desirable requirements for a private Docker Registry, the terms of service of DockerHub make this Registry not an option for the ElasTest Project. Once discarded DockerHub, Amazon EC2 Container Registry provides a good balance between costs, security and personalization, where the maintenance, backups and IAM can be configured easily without a major effort.

[29] Amazon EC2 Container Registry https://aws.amazon.com/ecr/?nc1=h_ls

The presented CI servers have the possibility for executing multiple kind of tests. Most of them use some kind of plugin that would launch or consume some other tool that execute the tests. Most of these plugins just use one tool and this tool is specialized in some kind of tests. For launching different types of tests, the user should configure different plugins and then develop or configure a way to integrate the results of all the plugins. This makes the procedure of testing quite tedious and little reusable. Also, the larger the system gets the harder this integration gets, the longer the test would take and the less information the tester gets as result.
ElasTest present 2 ways of addressing the difficulties that the presented CI servers have. First ElasTest is a whole platform that will focus on the test stage providing a whole set of functionalities to manage all the tests that can be applied to an application, moreover it focuses not only on application level but on the whole System in the Large. It provides a structured and complete interface to monitor and analyze the whole system and logs of the tests executed. The second solution that ElasTest presents is a way of integrating all its functionalities on the existing CI servers based on an ElasTest plugin, reducing the number of plugins needed for executing a complete suite of tests of different nature (functional, non-functional, performance, etc.), therefore the complexity of the configuration of the CI server would be reduced significantly. This second approach will be addressed with a PoC on the selected CI server (Jenkins), to prove the potential of ElasTest and the simplicity for having the already configured tests run in an ElasTest instance.

4.2 Non-functional Testing

In this section, we provide an overview of the most important tools concerning non-functional testing. Specifically, we divided them into the following two main categories that are of interest in ElasTest: application QoS testing and networking QoS testing. Below, there is a description of tools belonging to each category.

4.2.1 Application QoS Testing - Baseline and comparative analysis

In this section, we present the available tools for application QoS testing (Table 11). Specifically, we focus on performance and scalability testing that are two important objectives of ElasTest project. We present below a short description of the most relevant tools and finally a comparison among them in Table 12.

Name	URL	Brief description	License
JMeter	http://jmeter.apache.org/	The Apache JMeter application is open source software, a 100% pure Java application designed to load test functional behavior and measure performance. It was originally designed for testing Web Applications	OSS

		but has since expanded to other test functions.	
WebLOAD	http://www.radview.com/	WebLOAD is load testing tool, performance testing, stress test web applications.	Proprietary
LoadUI NG Pro	https://smartbear.com/product/ready-api/loadui/overview/	It is a load testing tool targeted mainly at web services (SOAP, REST).	Proprietary
Apica LoadTest	https://www.apicasystem.com/load-testing/	It is a load testing tool aimed to reveal how web applications behave under both normal and peak load conditions.	Proprietary
Appvance UTP	http://appvance.com/	Appvance UTP (Unified Test Platform) is a performance and load testing framework which allows to create load tests for a variety of application types, such as HTML5, AngularJS, Java Thick Client, Oracle Forms, Windows Clients, Citrix, JavaFX, mobile and more.	Proprietary
NeoLoad	https://www.neotys.com/neoload/overview	NeoLoad is a load and performance testing tool that realistically simulates user activity and monitors infrastructure behavior so the user can eliminate bottlenecks in all web and mobile applications.	Proprietary
WAPT	https://www.loadtestingtool.com/	WAPT is a load and stress testing tool that lets the user easily analyze the performance of web site.	Proprietary
Loadster	https://www.loadsterperformance.com/	Loadster is a full-featured load testing solution for websites, web apps, and web services.	Proprietary
LoadImpact	https://loadimpact.com	It allows for on-demand performance and capacity testing for enterprise applications.	Proprietary

Rational Performance Tester	http://www-03.ibm.com/software/products/en/performance	It is a tool for automated performance testing of web- and server-based applications from the Rational Software division of IBM.	Proprietary
Testing Anywhere	http://www.automationanywhere.com/Testing	It is a load test tool to assess web applications, controls and GUI front-ends.	Proprietary
OpenSTA	http://www.opensta.org/	It is a GUI-based web server benchmarking utility that can perform scripted HTTP and HTTPS heavy load tests with performance measurements.	OSS (GPLv2)
LoadStorm	https://loadstorm.com/	It is a load test for web and mobile application via the cloud. It uses scripts to simulate user behavior for up to a million virtual users.	Proprietary
CloudTest	https://www.soasta.com/load-testing/	CloudTest is a cloud-based testing application for load and performance tests of mobile and web applications.	Proprietary
HP LoadRunner	http://www8.hp.com/us/en/software-solutions/loadrunner-load-testing/	It is an extensible and easy-to-use framework for generating and running application load testing of Web 2.0 and mobile web applications.	Proprietary
SOASTA	https://www.soasta.com/load-testing/	It allows to create continuous flexible test to deliver scalable and high performing mobile and web applications	Proprietary
Blaze Meter	https://www.blazemeter.com/	It allows to configure and launch tests using any major open source performance testing tool	Proprietary

vPerformer	http://vperformer.findmysoft.com/	It can be used to assess the performance and scalability of web applications	Proprietary
The Grinder	http://grinder.sourceforge.net	It is a java based load testing tool. It can load test anything that has a Java API	OSS (LGPL v2.1)
LoadComplete	https://support.smartbear.com/viewarticle/78456/	LoadComplete is a load testing tool for creating and running automated load tests for web services. It will help to check web server's performance under a massive load, determine its robustness and estimate its scalability. LoadComplete creates load tests by recording actions the users perform over web pages and simulating these actions with dozens and hundreds of virtual users working simultaneously.	Proprietary
LoadView	https://www.loadviewtesting.com/scalability-testing/	As the target application is flooded with simultaneous users from the testing platform (using real browsers), LoadView allows to control the load curve (average response time, server load). It simulates globally distributed traffic, or local traffic all from one geographic location. Multiple cloud infrastructures are available including Google, Amazon, and Rackspace.	Proprietary
Pylot	https://qatestingtools.com/testing-tool/pylt	Pylot is a free open source tool for testing performance and scalability of web services. Pylot generates concurrent load (HTTP Requests), verifies server responses, and produces reports with metrics. Tests	GPL

	suites are executed and monitored from a GUI or shell/console.

Table 11. Application QoS testing tools

The most relevant tools for application QoS testing are: JMeter, LoadRunner, WebLOAD, Rational Performance Tester, The Grinder.
The analysis of these tools is based on the following characteristics:

- Response Time
- Throughput
- Hits per second, Request per seconds, Transaction per seconds
- Performance measurement with number of users
- Performance measurement under huge load
- CPU usage, Memory usage while testing in progress
- Network Usage - data sent and received
- Web server - Request and response per second

All these characteristics are common to the most relevant tools of the study. We will look at them in more detail with the aim of finding the differences between them.

4.2.1.1 *JMeter*

JMeter[30] is an open source performance and load testing tool, which is developed and maintained by Apache. JMeter is mostly used for load testing of web services and web application servers. Testing teams are also known to use it for functional testing of web services. It is compatible with Windows, Mac and all UNIX based systems. Reports generated are very limited and teams tend to use external plugins/tools for more detailed reporting. It supports the following protocols: Web - HTTP, HTTPS (Java, NodeJS, PHP, ASP.NET, …), SOAP / REST Webservices, FTP, Database via JDBC, LDAP, Message-oriented middleware (MOM) via JMS, Mail - SMTP, POP3 and IMAP, Native commands or shell scripts, TCP and Java Objects.
It has one of largest online community which shares helpful information with each other through different forums and software testing blogs.

4.2.1.2 *LoadRunner*

LoadRunner[31] is a commercial performance testing solution developed by HP. It has very advanced set of features which usually do not come built in with open-source or free tools.
LoadRunner can operate from Windows operating system. It is best known for its very detailed reports which help a lot in analyzing the performance issues. It supports the

[30] http://jmeter.apache.org/

[31] http://www8.hp.com/us/en/software-solutions/loadrunner-load-testing/

following protocols: .NET, Ajax (Click & Script), C Vuser, Citrix ICA, COM/DCOM, DNS, Flex, FTP, IMAP, Java over HTTP, Java Record Replay, Java Vuser, LDAP, MAPI, MMS (Media Player), MMS (Multimedia Messaging Service), Mobile Application – HTTP/HTML, ODBC, Oracle – 2 Tier, Oracle – Web, Oracle NCA, POP3, RDP, RTE, SAP, SAP GUI, SAP - Web, Siebel - Web, Silverlight, SMP, SMTP, TruClient – Firefox / IE, TruClient – Mobile Web, Web (HTTP/HTML), Web Services and Windows sockets.
HP provides dedicated support & knowledge base for LoadRunner to all registered customers. Other than that, one can also find plenty of articles and video tutorials online on HP LoadRunner.

4.2.1.3 *WebLOAD*

WebLOAD[32] is a commercial load testing, performance testing and stress testing tool for web applications developed by RadView Software. It Is compatible with Windows and Linux. It has a powerful analysis tool and customizable report. It supports the following protocols: HTTP/HTTPS (SSL, TLS), WebSocket, PUSH, AJAX, SOAP, HTML5, WebDAV and others. There is no solid community for webLOAD, but RadView offers documentation on its web and support to licensed customers.

4.2.1.4 *Rational Performance Tester*

Rational Performance Tester[33] is a performance testing solution developed by IBM. It is often used to test enterprise level applications like SAP, Oracle, etc. It can run on Windows, Mac and Linux AIX, offering detailed reports. It supports the protocols such as HTTP, SAP, TCP Socket, Citrix, Web Services (SOA), Siebel and TN3270. IBM provides dedicated support to its licensed customers. But other than that, availability of help material is quite little, if compared to other famous tools like JMeter or LoadRunner.

4.2.1.5 *The Grinder*

The Grinder[34] is a load testing tool described as a Java Load testing framework. It is open source and can run on Windows and Linux. It supports protocols such as HTTP, SOAP, REST web services, application servers (CORBA, RMI, JMS, EJBs), as well as custom protocols with hand-made plugins. The Grinder's documentation is very basic and hasn't a variety community resources.

[32] http://www.radview.com/

[33] http://www-03.ibm.com/software/products/en/performance

[34] http://grinder.sourceforge.net

	JMeter	LoadRunner	WebLOAD	Rational	Grinder
Multiplatform	+	-	+	+	+
Integration	+	+	+	+	+
Unlimited load generation	+	-	+	-	+
Executes JavaScript and render HTML	-	+	+	+	-
Great protocols support	-	+	-	-	-
Open source	+	-	-	-	+
Very Detailed Reports	-	+	+	+	-
Community	+	+	-	-	-
Support	-	+	+	+	+
Detailed Documentation	+	+	+	+	-
			+: Available	*, -: Not Available*	*, +/-: Partially available*
			Common	*Advantage*	*Disadvantage*

Table 12. Comparison of application QoS testing tools

4.2.2 Networking QoS Testing - Baseline and comparative analysis

We specify Networking QoS as a combination of metrics and policies. Metrics measure specific quantifiable attributes of the system components while policies dictate the behavior of the system components. Metrics can be further grouped into the following classifications: performance specifications and security levels, whereas policies are divided into level of service and management. Table 13 presents a list of networking QoS testing tools. The most relevant ones are: OpenDaylight, FloodLight, OpenStack Neutron. Below, there is a short description and a comparison among them (summarized in Table 14). Specifically, these tools have been compared according to the following characteristics: Metrics, Policies, Activity, License, and Management.

Name	URL	Brief description	License
ATOS SLA-Framework	https://github.com/Atos-FiwareOps/sla-framework	The SLA Framework is an implementation of an SLA lifecycle manager system, compliant with WS-Agreement. This project is part of FIWARE. The SLA Framework is a web, multi-platform application that allows to manage the whole lifecycle of service level agreements (from template creation to violation detection). It is a plugin-based decoupled component that can be adapted and extended to work on different platforms. It is an open source project, released under the Apache 2 license.	OSS (Apache 2.0)
OpenStack Neutron	https://docs.openstack.org/neutron/latest/	Neutron is an OpenStack project to provide “network connectivity as a service” between interface devices (e.g., vNICs) managed by other OpenStack services (e.g., nova). It implements the Neutron API.	OSS (Apache 2.0)
OpenDaylight	https://www.opendaylight.org	The OpenDaylight platform (ODL) provides a flexible common platform underpinning a wide breadth of applications and Use Cases.	Eclipse Public License - v 1.0
Floodlight	http://www.projectfloodlight.org/floodlight/	Floodlight is an OpenFlow controller (the "Floodlight Controller") and a collection of applications built on top of the Floodlight Controller.	OSS (Apache 2.0)
Open vSwitch	http://openvswitch.org/	Open vSwitch is a production quality, multilayer virtual switch. It is designed to enable massive network automation through programmatic extension, while still supporting standard management interfaces and protocols (e.g. NetFlow, sFlow, IPFIX, RSPAN, CLI, LACP, 802.1ag).	OSS (Apache 2.0 license)
Indigo Virtual Switch	http://www.projectfloodlight.org/indigo-virtual-switch/	Indigo Virtual Switch (IVS) is an open source virtual switch for Linux compatible with the KVM hypervisor and leveraging the Open vSwitch kernel module for packet forwarding. IVS is built on the Indigo Framework and leverages LoxiGen generated	Eclipse Public License - v 1.0

	code (loci) to handle OpenFlow messages.

Table 13. Networking QoS testing tools

4.2.2.1 *OpenDaylight*

The OpenDaylight platform (ODL)[35] provides a common foundation and a robust array of services to enable a wide breadth of applications and use cases. ODL delivers the benefits of SDN and NFV to carriers, enterprises, research institutions, and other organizations such as cities and metropolitan areas, dynamically optimizing the network based on load and state. This is the most common carrier use case as it optimizes the network using the near-real-time state of traffic, topology and equipment.

Metrics: OpenDaylight provides a Monitoring-as-a-Service feature called Cardinal[36]. Cardinal enables OpenDaylight and the underlying software defined network to be remotely monitored by deployed Network Management Systems (NMS) or Analytics suite.

Policies: It readily supports proprietary and extended network services, including path computation, resource management, analytic for both virtual and physical domains. Opendaylight applies policies through a Group Based Policy project[37] which allows users to declare policies in a declarative way.

Activity: The OpenDaylight foundation promotes and advances the global development, distribution and adoption of the OpenDaylight (ODL) open source SDN platform. Founded in 2013, they maintain an independent governance and infrastructure to help ensure that ODL's development can occur in a neutral environment.

License: Each of the separate OpenDaylight projects designated in the development environment as a Program is made available under the Eclipse Public License (EPL)-1.0 as a separate Program[38].

Management: It supports Intent-based Northbound (i.e., Network Application to Controller) interfaces exposing SDN capabilities to diverse network applications, while abstracting the underlying infrastructure details.

4.2.2.2 *FloodLight*

Floodlight[39] is an Open SDN Controller that works with virtual and physical switches that speak with the OpenFlow protocol.

OpenFlow is an open standard that enables researchers to run experimental protocols in the research networks we use every day. OpenFlow is added as a feature to commercial Ethernet switches, routers and wireless access points – and provides a

[35] https://www.opendaylight.org

[36] http://docs.opendaylight.org/en/stable-boron/user-guide/cardinal -opendaylight-monitoring-as-a-service.html

[37] https://wiki.opendaylight.org/images/9/90/Gbp-lithium-user-guide.pdf

[38] https://www.opendaylight.org/licensing

[39] http://www.projectfloodlight.org/

standardized hook to allow researchers to run experiments, without requiring vendors to expose the internal workings of their network devices. Floodlight works with physical and virtualized switches[40].
Metrics: A limited monitoring capability is provided through the API.
Policies: FloodLight currently features two reactive packet forwarding applications: i) within an OpenFlow cluster; ii) between two OpenFlow clusters with a non-openflow cluster between them.
Floodlight does not support forwarding for topologies that have any kind of cycles between clusters.
Activity: FloodLight provides from time to time some source code updates whereas the latest release of FloodLight is already more than two years old, more specific, v1.2 was released in March 2016.
Licence: The FloodLight controller is licensed under the Apache License 2.0 and is available on GitHub.
Management: For controlling the FloodLight tool it can be used a separated GUI or the RESTful API directly.

4.2.2.3 *OpenStack Neutron*

The OpenStack Networking service provides an API that allows users to set up and define network connectivity and addressing in the cloud. The project code-name for Networking services is neutron. OpenStack Networking handles the creation and management of a virtual networking infrastructure, including networks, switches, subnets, and routers for devices managed by the OpenStack Compute service (nova). Advanced services such as firewalls or virtual private networks (VPNs) can also be used. OpenStack Networking consists of the neutron-server, a database for persistent storage, and any number of plug-in agents, which provide other services such as interfacing with native Linux networking mechanisms, external devices, or SDN controllers.
OpenStack Networking is entirely standalone and can be deployed to a dedicated host. If the deployment uses a controller host to run centralized compute components, the networking server can be deployed to that specific host instead.
As part of the OpenStack project, it Integrates with Keystone (identity), Nova (compute) and Horizon (dashboard)[41].
Metrics: There is a long list of monitoring tools that integrate with OpenStack. Among them, there is also Monasca. Monasca is an open-source multi-tenant, highly scalable, performant, fault-tolerant monitoring-as-a-service solution that integrates with OpenStack. It uses a REST API for high-speed metrics processing and querying and has a streaming alarm engine and notification engine.
Policies: It supports The Group Based Policy (GBP) extension that introduces a declarative policy driven framework for networking in OpenStack. The GBP abstractions allow application administrators to express their networking requirements using group

[40] https://floodlight.atlassian.net/wiki/spaces/floodlightcontroller/pages/1343519/Compatible+Switches

[41] https://docs.openstack.org/developer/neutron/feature_classification/general_feature_support_matrix.html

and policy abstractions, with the specifics of policy enforcement and implementation left to the underlying policy driver. It consists of the following:

- Groups - a group that represents a collection of network endpoints and fully describes their properties. Everything in the same group must be treated the same way (that is it has the same policy).
- Policy rule sets - Rule sets describe secure connectivity between Groups. Rule sets may imply switching or routing behaviors, but they offer a simple way to describe how sets of machines can communicate in non-networking terms.
- Policy layering - policies to be layered based on different roles in an organization. E.g. application owners can specify the policy pertaining to an application, while infrastructure owners can prescribe security requirements and both policies can coexist.
- Network Service - supports a redirect operation that complex network service chains and graphs can easily abstract and consume.

Automation and security are easy to set up, because by simply becoming a member of a group, a virtual machine inherits all of its policies, allowing developers to easily automate scaling up and down[42].

Activity: OpenStack is a set of software tools for building and managing cloud computing platforms for public and private clouds.

License: The OpenStack Neutron component is part of the OpenStack project and follows the Apache License 2.0.

Management: OpenStack Neutron is partially controllable via the Horizon dashboard which uses the RESTful API exposed by Neutron directly. Some more specific operations must be issued via the command line interface which is written in python. In addition, there are several clients and libraries available which allows to talk to Neutron within the own application written in python, Java or other languages.

	OpenDaylight	FloodLight	OpenStack Neutron
Metrics	Analytics for both physical and through a mobility as a service feature	Limited API support	Multiple monitoring tools among which Monasca
Policies	Group Based Policy built as an Intent System	Limited reactive packet forwarding	Group Based Policy extension
Activity	High	Low	High
License	Eclise Public License 1.0	Apache 2.0	Apache 2.0

[42] https://wiki.openstack.org/wiki/GroupBasedPolicy

Management	REST API	REST API, GUI	REST API, GUI, SDKs

Table 14. Comparison of networking QoS testing tools

4.2.3 Progress within ElasTest

ElasTest will allow performance testing combined with scalability thanks to ElasTest configuration management mechanisms, and the Test Support Services.
ElasTest will make progress beyond SotA by allowing testers and developers to assess their SiL by running different configurations and comparing the results among them. This will enable teams to choose the most appropriate scalability approach. Current tools for load testing don't enable running the SuT under different conditions and don't enable testers to compare results from different configurations. ElasTest will provide some sort of performance testing. We will use these tools to gather ideas for the implementation/features/user interface. The most open source tools are desktop tools that need to be installed to be used. ElasTest allows integrating those tools. For example, ElasTest could execute JMeter performance tests and show the results in its web interface. This work is being reinforced by our collaboration with H2020 STAMP Project[43] to bring generated test configurations to ElasTest.

The ElasTest Platform Manager (EPM) will integrate some of these QoS tools, and also it will enable the definition of QoS Policies that are agnostic from the underlying infrastructure and tool in use.

4.3 Security testing

4.3.1 Baseline and comparative analysis

Table 15 shows a list of security testing tools. The most relevant tools dealing with security testing are: OWASP ZAP, Burp Suite Professional, Metasploit, WireShark. We provide below a brief description of them.

Name	URL	Short Description	License
Peach	https://www.peach.tech/	Protocol-aware fuzzer. The Peach fuzzer allows the tester to define grammars of network protocols that will be used by the framework to perform protocol-level fuzzing.	OSS, a commercial version also exists

[43] https://www.stamp-project.eu/view/main/

Mozilla Peach	https://wiki.mozilla.org/Security/Fuzzing/Peach	Protocol-aware fuzzer. A fork by Mozilla of the Peach fuzzer.	OSS
NetZob	http://doc.netzob.org/en/latest/index.html	Protocol reverse-engineering framework. Netzob allows to recover information from network protocols for which no specification is available.	OSS
Wfuzz	https://github.com/xmendez/wfuzz	Web application bruteforcer. It enables to find weak credentials in web applications.	OSS
OWASP ZAP	https://github.com/zaproxy/zaproxy	Vulnerability scanner for Web applications.	OSS
SQLmap	https://github.com/sqlmapproject/sqlmap	Vulnerability scanner for Web applications.	OSS
Burp	https://portswigger.net/burp/	Vulnerability scanner for Web applications.	Free (not OSS)
w3af	https://github.com/andresriancho/w3af	Vulnerability scanner for Web applications.	OSS
Arachni	https://github.com/Arachni/arachni	Vulnerability scanner for Web applications.	OSS
SkipFish	https://github.com/spinkham/skipfish	Vulnerability scanner for Web applications.	OSS
Vega	https://subgraph.com/vega/	Vulnerability scanner for Web applications.	OSS
Fortify WebInspect	http://www8.hp.com/us/en/software-solutions/webinspect-dynamic-analysis-dast/	Vulnerability scanner for Web applications. It applies automated dynamic application security testing that mimics real-world hacking techniques and attacks and provides comprehensive dynamic analysis of complex web applications and services.	Proprietary
Netsparker	https://www.netsparker.com/	Vulnerability scanner for Web applications. It finds and reports web application vulnerabilities such as SQL Injection and Cross-site Scripting (XSS) on all types of web applications.	Proprietary

Nmap	https://nmap.org/	A tool for network discovery and security auditing, useful for discover what hosts are available on the network, what services those hosts are offering, what operating systems they are running, and other characteristics.	OSS
CharlesProxy	https://www.charlesproxy.com/	An HTTP proxy / HTTP monitor / Reverse Proxy that enables a developer to view all of the HTTP and SSL / HTTPS traffic between their machine and the Internet.	Proprietary
WireShark	https://www.wireshark.org/	Network protocol analyzer. Wireshark embeds parsers for a very large number of network protocols.	OSS
SSO Scan	http://ssoscan.org/	Tool for detecting vulnerabilities in Single Sign On implementations.	private
MITM Proxy	https://mitmproxy.org/	It is a source interactive HTTPS proxy. It allows for debugging, testing, privacy measurements, and penetration testing. It can be used to intercept, inspect, modify and replay HTTPtraffic.	OSS
Metasploit	https://www.metasploit.com/	Metasploit helps security teams to verify vulnerabilities, manage security assessments, and improve security awareness.	OSS (also commercial version)

Table 15. Security testing tools

4.3.1.1 *OWASP ZAP*

OWASP ZAP [44] One of the most-popular free and open-source web vulnerability scanners. It contains a lot of functionality related to security testing, in particular with respect to finding vulnerabilities in Web applications. It is a good representative of the myriad of existing web vulnerability scanners and it helps in identifying the most-common security vulnerabilities (e.g., SQL injection, cross-site scripting etc.). ZAP also come with a Python and REST API that allows another application to programmatically control it. The following are the important modules within ZAP:

1. Man-in-the-Middle (MITM) proxy: A module for intercepting HTTP and HTTP communications between a Web client and a Web server;
2. Spider: A module for automatically crawling a web application for achieving coverage while testing;
3. Passive Scan: This module checks the HTTP requests and responses collected by ZAP and checks it for security vulnerabilities without modifying its contents (hence it is safe to use);
4. Active Scan: This module sends HTTP requests to the web application (which is under test) that attacks the Web application and the corresponding HTTP responses are analyzed for understanding whether the attacks worked (and thereby detecting the presence of vulnerabilities).

4.3.1.2 *Burp Suite Professional*

Burp Suite Professional[45] is a commercial web application security testing tool (there is also a free version available) that includes a man-in-the-middle proxy, vulnerability scanner and other tools for manual penetration testing. Other than the common Web application weaknesses, Burp Suite Professional also supports the detecting of other vulnerabilities such as blind XML External Entity (XXE) injection. The most widely used modules of Burp are similar to the ones explained for ZAP (i.e. Spider, Passive/Active Scan etc.).

4.3.1.3 *Metasploit*

Metasploit[46] Framework is a software platform for developing, testing, and executing exploits. It can be used to create security testing tools and exploit modules and also as a penetration testing system. Metasploit also offers a payload database, allowing the pen tester to mix and match exploit code and objectives. Written using Ruby, the tool has about 500 modules, including hundreds of remote exploits that can be targeted against various releases of Windows, Linux, BSD, Unix, and the Mac OS. The development of the Metasploit framework is supported by both the open source community and the company Rapid7. Metasploit can be used to test web applications, networks, servers etc.

[44] https://github.com/zaproxy/zaproxy

[45] https://portswigger.net/burp/

[46] https://www.metasploit.com/

4.3.1.4 *WireShark*

Although WireShark[47] by itself is not a security testing tool, it is the most widely-used network protocol analyzer. It allows to see what's happening on the network at a microscopic level and it is the de facto standard across many commercial and non-profit enterprises, government agencies, and educational institutions. It allows for: i) deep inspection of lots of protocols; ii) Live capture and offline analysis; iii) Standard three-pane packet browser; iv) Multi-platform: Runs on Windows, Linux, macOS, Solaris, FreeBSD, NetBSD, and many others; v) Captured network data can be browsed via a GUI, or via the TTY-mode TShark utility; vi) Rich VoIP analysis; vii) Read/write a wide set of different capture file formats; viii) Capture files compressed with gzip can be decompressed on the fly; ix) Live data can be read from Ethernet, IEEE 802.11, PPP/HDLC, ATM, Bluetooth, USB, Token Ring, Frame Relay, FDDI, and others (depending on your platform); x) Decryption support for the most common security protocols; xi) Coloring rules can be applied to the packet list for quick, intuitive analysis; xii) Output can be exported to XML, PostScript®, CSV, or plain text.

Table 16 summarizes the important features available in each of the tools.

	OWASP ZAP	Burp Suite Pro.	Metasploit	WireShark
Spidering feature available?	Yes	Yes	No	No
Passive Scan feature available?	Yes	Yes	Yes	No
Active Scan feature available?	Yes	Yes	Yes	No
MitM feature available?	Yes	Yes	No	Yes
Extendable using plugins?	Yes	Yes	Yes	Yes
Remotely controllable by external applications?	Yes	No	No	Yes
Type of System	Web Applications	Web Applications	Web Applications, Servers, Network Devices	Web Applications, Servers, Network Devices

[47] https://www.wireshark.org/

under Tests supported				
Free or Commercial	Free	Both free and commercial versions available	Both free and commercial versions available	Free

Table 16. Comparison of security testing tools

4.3.2 Progress within ElasTest

The Security Services component within ElasTest, also known as the ElasTest Security Service (ESS in short) enables security testing of cloud-based Web applications. ESS advances the state-of-the-art of security testing in the following ways. First, ESS advances the passive scan feature. Note that, various state-of-the-art tools (e.g., OWAZP ZAP and Brup Suite Professional) also support the passive scan feature. However, ESS not only contains the passive scanning modules within the state-of-the art tools like OWAZP ZAP but also has additional modules such as automatic identification of sensitive HTTP Cookies that are not protected from cross-origin attacks (this is done by identifying cookies that are missing the SameSite cookie attribute). Second, the active scan option within ESS supports the black-box detection of cross-origin attacks in Web applications (e.g., Login Oracle attack) that abuse authorization. Other state-of-the-art tools such as Burp Suite Professional do not have this option. Third, ESS supports the automatic detection of replay attacks using a novel technique inspired from program slicing. Currently, available tools that support the detection of replay attacks either use ad-hoc techniques or lacks automation. Other than this, ESS also advances the state-of-the-art in crawling Web applications (sometimes referred to as spidering). ESS introduces a Selenium-based crawler and an automatic user interaction automation technique to efficiently crawl a Web application. This crawling module is combined with the active and passive security scan options within OWASP ZAP with the help of the OWASP ZAP API to test a Web application for common Web applications weaknesses.

4.4 Monitoring

4.4.1 Baseline and comparative analysis

A monitoring framework/system comprises of set of technology components (hardware and/or software) that enables elements in an IT ecosystem (users, admins, other hardware/software modules) to monitor the status and performance of a specific software/hardware component under study. The primary goal of such a service is to enable gathering of relevant monitored data streams and later making the stored entries available to other consumers for further analysis such as visualization of trends over period of time, SLA validation, audit workflows, etc. Typically, in any monitoring system, data retention is governed via a policy that is in line with the natural EoL for the specific data being retained.

Typically, any modern monitoring system supports following data/metric lifecycle stages depicted in Figure 17.

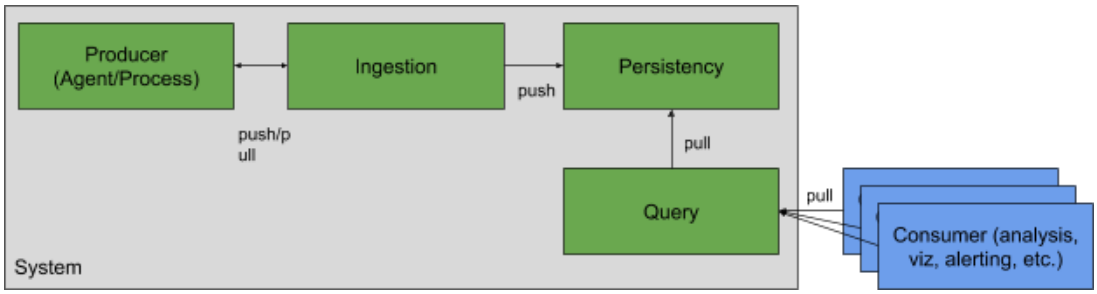

Figure 17. Logical architecture of a monitoring system

In particular, a monitoring system has the following components: i) Agent: a process that reports metrics to an input interface; ii) Input interface, metric consumption namely an interface that ingests either push or pull; iii) Metric persistence; iv) Metric query engine; v) Query interface.

As shown in the diagram above, a monitoring system is not the system that provides the capabilities of alerting, fault detection, system tracing or metric analysis. These capabilities are built by higher level systems that are consumers of the metrics consumed through the query interface.

Table 17 shows the available monitoring tools. Among them, the most relevant ones are: Prometheus, InfluxData TICK, Zabbix, and Datalog.

In the following, we analyze them showing how they map the main components of a logical architecture of a monitoring system showed in Figure 17 that are: Producer, Ingestion, Persistency, Query and Consumer. Finally, Table 18 shows a comparison of these tools.

Name	URL	Short Description	License
cAdvisor	https://github.com/google/cadvisor	It analyzes resource usage and performance characteristics of running containers.	Apache License, Version 2.0
Prometheus	https://prometheus.io/	It is a monitoring system based on a pull approach with a time series database	Apache License, Version 2.0
Agentless System Crawler	https://developer.ibm.com/code/open/projects/agentless-system-crawler/	It is a tool to crawl systems like crawlers for the web	Apache License, Version 2.0
collectd	https://collectd.org/	It collects system and application performance metrics periodically and provides mechanisms to store the values in a variety of ways	MIT and GPL 2
fluentd	http://www.fluentd.org/	It is an open source data collector for unified logging layer	Apache License, Version 2.0

SNAP	https://snap-telemetry.io/	It is an open telemetry framework that collects, processes, and publishes telemetry data	Apache License, Version 2.0
Nagios Core	https://www.nagios.org/	It is an IT infrastructure monitoring and alerting	OSS
iCINGA	https://www.icinga.com/	It is a monitoring system which checks the availability of your network resources, notifies users of outages, and generates performance data for reporting	GPL2
Riemann	http://riemann.io/	It is a distributed system monitoring framework using stream processing and a powerful, concise, extensible syntax	EPL 1.0
Ganglia	http://ganglia.info/	It is a scalable distributed monitoring system for HPC, it is targeted at cluster of grids in a federated fashion	BSD
ZABBIX	https://www.zabbix.com/	Zabbix is open source enterprise monitoring tool to track the availability and performance of IT infrastructure components	OSS
sysdig	https://github.com/draios/sysdig	sysdig is a system level exploration tool that allows aggregation, filtering and analysis and comes with a containers view to allow useful correlation between different traces	GPL2
SYSDIG-FALCO	https://sysdig.com/opensource/falco/	It is a behavioral activity monitor designed to detect anomalous activity in your applications. It supports combination of snort, ossec and strace	GPL2
OpenTracing	http://opentracing.io/	OpenTracing offers consistent, expressive, vendor-neutral APIs for popular platforms. OpenTracing makes it easy	OSS

		for developers to add (or switch) tracing implementations by means of simple configuration changes. OpenTracing also offers a language for OSS instrumentation and platform-specific tracing helper libraries	
Cacti	https://www.cacti.net/	Complete frontend to RRDTool	GPL2
InfluxData TICK	https://www.influxdata.com/time-series-platform/	Build monitoring and analytics applications	MIT Licensed
Pingdom	https://www.pingdom.com/	Pingdom offers a simple way to monitor the availability and performance of websites, servers and web applications	Apache License, Version 2.0
sensu	https://sensu.io/	Open source monitoring tool for distributed applications and various system services. Sensu is written in Ruby that uses RabbitMQ to handle messages	Apache License, Version 2.0
scout	https://scoutapp.com/	Scout is a simple hosted server and application monitoring service. Main feature is ease installation and configuration as well as possibility to create plugins to extend Scout. Scout allows default alerts to help administrators understand how the application is behaving under various loads	Apache License, Version 2.0
datadog	https://www.datadoghq.com/	It allows for quickly search, filter, and log analysis. It enables events correlation as well as generation and upload of JSON-formatted dashboards	MIT and GPL 2

Oracle CQL	https://docs.oracle.com/cd/E12839_01/apirefs.1111/e12048/loe.htm	This tool introduces a query language based on SQL with added constructs that support streaming data, which allows expressing queries on data streams to perform complex event processing (CEP) using Oracle CEP	Apache License, Version 2.0

Table 17. Monitoring tools

Prometheus

Prometheus[48] is a monitoring and alerting framework originally built at SoundCloud. It uses a pull model for data ingestion. Since 2016, Prometheus code is part of Cloud Native Computing Foundation making it a framework of choice for monitoring microservices, especially deployments running as docker containers and managed via Kubernetes. The main features are: i) a multi-dimensional data model; ii) a flexible query language; iii) no reliance on distributed storage; single server nodes are autonomous; iv) time series collection happens via a pull model over HTTP; v) pushing time series is supported via an intermediary gateway; vi) targets are discovered via service discovery or static configuration; vii) multiple modes of graphing and dashboarding support.
Since data collection is pull based, Prometheus servers periodically poll for new samples from agents. In case of short lived processes, Prometheus supports such processes to push their metric to a Pushgateway, from where it reads any data at a set periodicity.

[48] https://prometheus.io/

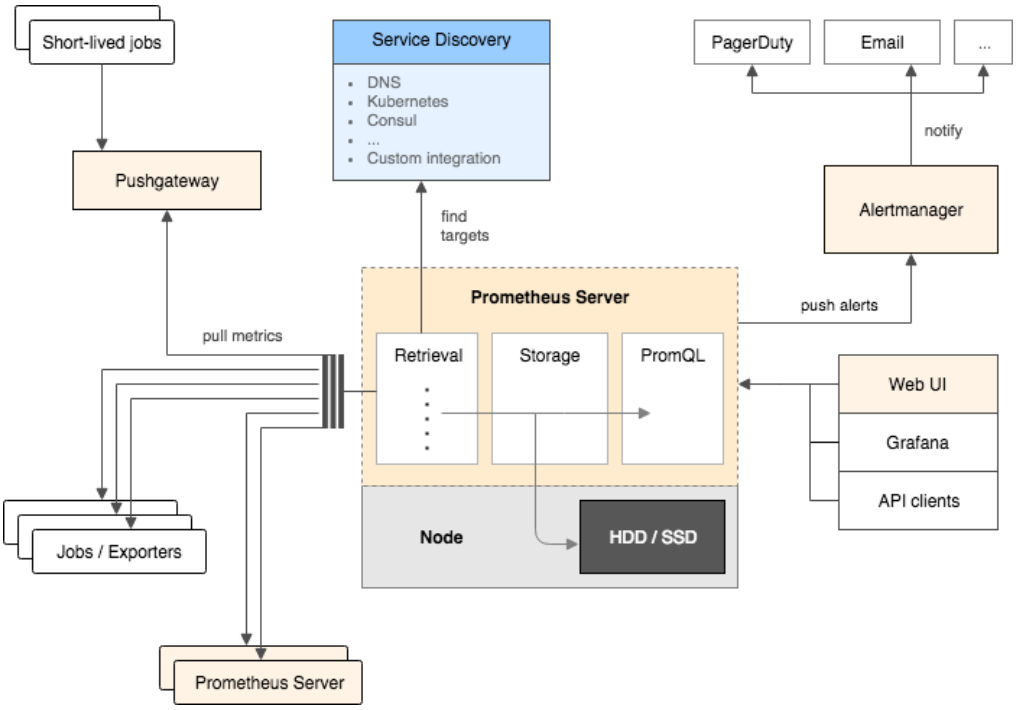

Figure 18. Prometheus architecture[49]

Architecture: As shown in Figure 18 the main components of Prometheus include server that polls and collects data periodically from exporters which are developed using numerous client libraries for instrumenting client code. It also includes Alertmanager that processes the polled data and based on rules can trigger different actions. PromQL is an expressive query language that allows external entities to fetch stored data from Prometheus. Such queries can be used to enable visualization using well known tools such as Grafana. We show below how Prometheus maps to the logical architecture of a monitoring system shown in Figure 17.

Producer: Exporters in Prometheus are the metrics producers. Exporters make the collected metrics available to Prometheus for pulling at a configured periodicity. Prometheus comes with a wide variety of exporters covering a large number of systems and services[50].

Ingestion: The Prometheus server is responsible for data gathering and ingestion. It pulls data from all exporters and Pushgateway for short lived processes.

Persistency: Prometheus uses a self-contained local storage for data persistency. The design choice here was made to make Prometheus self-contained instead of dependent on external storage subsystems thus improving reliability - its main goal. The indexes are managed in LevelDB, a fast key-value store developed at Google. The bulk data is stored in fixed sized chunks of 1KB each. Chunks are stored as files in the local storage. On disk, the chunks of same metric series are collated in a single file. Currently used chunks as well as a set threshold of recently used chunks are maintained in working memory for faster access.

[49] src: https://prometheus.io/docs/introduction/overview/

[50] https://prometheus.io/docs/instrumenting/exporters/

Query: Prometheus provides a functional expression language that lets the user select and aggregate time series data in real time. The result of the query can be visualization engines, other applications, or Prometheus dashboard.
***Consumer*:** The consumer of Prometheus via its functional query interface is Prometheus dashboard itself. Visualization engines such as Grafana, or other systems via HTTP API calls are also supported.

4.4.1.1 *InfluxData TICK*

InfluxData[51] helps to build monitoring, analytics and IoT applications. InfluxData offers a non-complex and scalable solution for time-based data applications (Figure 19).

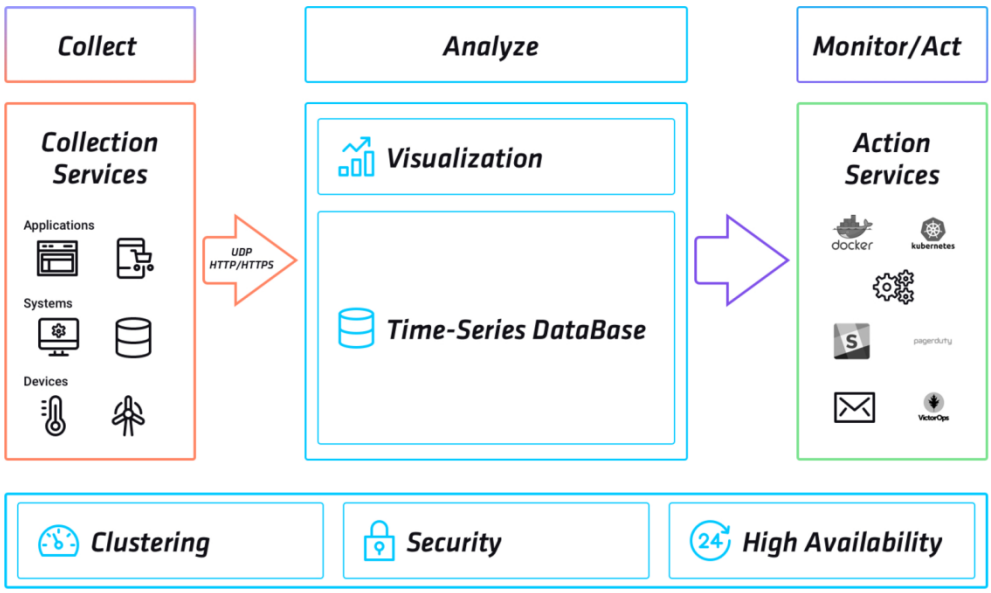

Figure 19. InfluxData[52]

Architecture: InfluxData provides several tools to get metrics and events data from different devices, sensors, systems, machines, containers and applications. TICK includes four different open source projects from InfluxData: Telegraf, InfluxDB, Chronograf and Kapacitor.
All the data collection services are built from the Telegraf service. Once the data is collected it needs to be stored. To do that InfluxData is using InfluxDB as a back-end database. All the data stored in InfluxDB can be visualized by using Chronograf or Grafana. Finally, Kapacitor offers the possibility to plug in custom logic or user defined functions to process alerts, match metrics for patterns, compute statistics, scale containers and basically anything that you can program.
Producer: TICK uses Telegraf as an agent to report metrics. Telegraf has output plugins to send metrics to different datastores, services and messaging queues including InfluxDB, Graphite, OpenTSDB, Datadog, Librato, Kafka, MQTT, NSQ, and many others.

[51] https://www.influxdata.com/products/open-source/

[52] src: https://www.influxdata.com/products/

Ingestion: Telegraf is also used to collect metrics. Telegraf has plugins or integrations to source different metrics directly from the systems it's running on, pull metrics from a third-party API or using services like Kafka.
Persistency: TICK uses InfluxDB as a back-end data storage. InfluxDB is a time series database so that every data point saved into the database will contain a "time" column that links that data point to a timestamp.
Query: InfluxDB uses its own DSL called InfluxQL. InfluxQL is a SQL like language and includes the standard expressions such as SELECT, WHERE, GROUP BY and also different functions for exploring the data like COUNT, MIN, MAX, MEDIAN and DERIVATIVE.
Consumer: By default, InfluxDB provides support via graphical user interface, command-line interface access, and HTTP API for data queries.

4.4.1.2 *Zabbix*

Zabbix[53] is a monitoring system that is designed to monitor network-based services. It is a monitoring system that has a key element of alert reporting within.
Architecture: The architecture of Zabbix is a traditional architecture with a main service that is backed by a database. In order to scale such a deployment, Zabbix can be configured to have child Zabbix servers attached to a parent server. Alternatively, Zabbix provides a proxy to the server that can be instantiated many times across hosts and act as front-ends to the Zabbix service. Scaling the persistency layer would follow how typical SQL deployments are scaled. In order to inspect the data ingested, Zabbix provides a web-based user interface, which requires a direct connection to the DB.
Producer: The producer in the Zabbix system is the Zabbix Agent. SNMP polls and events (traps) from supporting infrastructure components can also be used to supply metrics to the Zabbix server. Zabbix can monitor physical and virtual (VM) hosts and can also monitor containers deployed upon those hosts[54].
Ingestion: The Zabbix server is the entity responsible for ingesting metrics that it retrieves (pull) from Zabbix agents and then persisting this data to the Zabbix DB. Other than this, the Zabbix Server generates alerts based on threshold-based rules.
Persistency: Zabbix uses traditional SQL data DBMSes. It supports MySQL, PostgreSQL, SQLite, Oracle or IBM DB2 to store data.
Query: Zabbix Server API allows the retrieval and modification of Zabbix configuration and allows access to historical data. It uses the JSON-RPC 2.0 protocol[55]. Part of this API is to allow the querying of the collected historical data. The API can be exposed to consumer services that want to build upon the collected data.
Consumer: The default supplied consumer of the Zabbix System is the Zabbix user interface that allows collected metrics and alerts to be visualized but not interacted with dynamically.

[53] http://www.zabbix.com
[54] https://github.com/monitoringartist/Zabbix-Docker-Monitoring
[55] http://www.jsonrpc.org/specification

4.4.1.3 *Datadog*

Datadog[56] is a hosted monitoring service that allows metrics to be sent into a hosted monitoring service and the data points analyzed later. Datadog is a commercial provider where the agents are available as open source codebase but the core of the platform remains proprietary. Datadog seamlessly integrates both metrics and events. If a supported agent is used, then the platform supports automatic generation of graphs showing error rates, latency percentiles. It also supports error tracing by performing correlated analysis between application logs, metrics and events.

Architecture: Datadog stores all data in Amazon S3 as a replacement for HDFS. The processing of data points is done as an elastic map-reduce cluster in AWS, and uses Apache Pig, Spark and Python Luigi workflow management framework to analyze large amount of data streams. As a commercial service, the length of time a metric is preserved depends on the purchased plan and it typically varies from 1 day to 15 months.

Subsequent heading maps the Datadog capabilities along common dimension outlined earlier.

Producer: Datadog uses agent to send metrics from users' infrastructure into the hosted service. The job of an agent is to collect and send metrics and events from customers' environment faithfully. An agent code is made available as open source component.

Ingestion: Datadog agent itself sends the collected metrics to Datadog via a load balancer endpoint in AWS.

Persistency: Datadog uses Amazon S3 service for persistency in order to overcome the single cluster limitation of HDFS and the language dependency with Java.

Query: Datadog allows query in both time and space, they do automatic aggregation to downsample the data to bring it to a reasonable count to display in the browser. The Query API is accessible to the users if they are interested in using the data to build over the top applications.

Consumer: The main consumer is the Datadog web UI which allows the graphs and alerts to be visualized.

	Prometheus	InfluxData TICK	Zabbix	Datadog
OpenSource	Yes	Yes	Yes	No (only agents)
Data ingestion model	Push/Pull	Push by agents, Telegraf allows both pull/push model	Pull from agents	Push by agents
Persistence Layer	Append only file per time series	Log-structured merge tree with write ahead logs shared by time	Relational DBs	Amazon S3

[56] https://www.datadoghq.com/

Alerting support	Yes	Yes	Yes	Yes
Query Capability	Functional query language PromQL for live querying	SQL like DSL called InfluxQL	Full SQL query over JSON-RPC	Limited filters and group by over HTTP
3rd party integrations	Many	Limited	Many	Many

Table 18. Comparison of monitoring tools

4.4.2 Progress within ElasTest

While most monitoring systems are designed for either system metrics or logs, EMP supports both logs and metrics. In doing so, it allows similar set of operations on fields or tags associated with log data or system metrics. Of course, the claim is valid as long as the operation is meaningful for the operand fields. EMP is pushing the SoTA by incorporating both push/pull mechanisms. In contrast, most competing systems are either push or pull based. EMP has provisions for on-line metric processing along the way until the data is persisted. EMP allows both immediate processing of critical streams for timely interventions and longer analytic with older data sets. This makes it appropriate for monitoring of critical infrastructures as well as historical analytic. EMP query engine will allow correlated query with multiple data streams allowing comprehensive analysis towards fault tracing especially with micro-services based distributed large-scale services. The above listed features highlight some of the aspects beyond SoTA being researched in EMP.

4.5 GUI Automation and Impersonation

In this section, we describe GUI automation and impersonation tools. We divide them into two main categories: i) general GUI test automation tools presented in Section 4.5.1; and ii) device emulation tools specifically referring to emulation of IoT devices in order to realize and test IoT applications that are addressed into ElasTest (Section 4.5.2).

4.5.1 GUI automation - Baseline and comparative analysis

Table 19 shows a list of GUI automation tools. In the absence of unified criteria to choose the most popular tools in the developer community, concerning GUI automation, we will select a group of tools, based on the type of license, the features they provide and the type of graphical applications they test. They are: Selenium, Appium, Nightwatch.js, qTest Explorer, TestComplete. For this study, we have taken low-level and high-level tools. The low-level tools provide the API to perform the automated test and set up the test scenarios, and the high-level tools use this API to build and provide high-level functionalities like a codeless automated testing, configuring test environment through forms and so on. In the following we provide a comparison of these tools.

Name	URL	Short Description	License
Selenium	http://www.seleniumhq.org/	Selenium is a testing framework for web applications. It has 3 components: i) Selenim IDE: Firefox plugin that provides record/playback (R&P) capabilities; ii) Selenium WebDriver: Implementation of the WebDriver API (https://w3c.github.io/webdriver/webdriver-spec.html) for controlling browsers in different programming languages (e.g. Java, JavaScript, Python, Ruby, among others); iii) Selenium Grid: allows running WebDriver tests on different machines (distributed test execution).	OSS
Appium	http://appium.io/	Appium is an open source framework that allows automating on mobile devices. It allows test automation for native, hybrid and mobile web apps, tested on simulators (iOS), emulators (Android), and real devices (iOS, Android, Windows).	OSS
Nightwatchjs	http://nightwatchjs.org/	Nightwatch.js is an easy to use Node.js based End-to-End (E2E) testing solution for browser-based apps and websites. It uses the powerful W3C WebDriver API to perform commands and assertions on DOM elements.	OSS (MIT)
Ascentialtest	http://www.zeenyx.com/AscentialTest.html	It allows to build automated test steps by interacting with graphical representations of the application under test.	Proprietary
AutoIt	https://www.autoitscript.com/site/autoit/	AutoIt v3 is a freeware BASIC-like scripting language designed for automating the Windows GUI and general scripting.	Freeware
Eggplant Functional	https://www.testplant.com/	It is a black-box GUI test automation tool. Its approach uses image matching technology as opposed to looking to the object-level of the application being tested.	Proprietary

iMacros	http://imacros.net/	iMacros is an extension for the Mozilla Firefox, Google Chrome, and Internet Explorer web browsers which adds record and replay functionality similar to that found in web testing and form filler software.	Proprietary
Linux Desktop Testing Project	https://github.com/ldtp	The Linux Desktop Testing Project (LDTP) is an open source testing tool that uses computer assistive technology to automate GUIs.	OSS (LGPLv2.1)
Maveryx	http://www.maveryx.com/	Maveryx is an automated functional, graphical user interface (GUI), and regression test tool for Java and Android applications.	OSS (GPLv2)
Oracle Application Testing Suite	http://www.oracle.com/technetwork/oem/app-test/index.html	It is an integrated testing solution for Web applications, Web Services, packaged Oracle Applications and Oracle Databases. It is comprised of the following tightly integrated products: i) Oracle Functional Testing - automated functional and regression testing of web applications; ii) Oracle Functional Testing Suite for Oracle Applications - functional and regression testing of Oracle packaged applications; iii) Oracle Load Testing - scalability, performance and load testing of web applications; iv) Oracle Load Testing Suite for Oracle Applications - scalability, performance and load testing of Oracle packaged applications: v) Oracle Test Manager - test process management, including test requirements management, test management, test execution and defect tracking.	Proprietary
QF-Test	https://www.qfs.de/en.html	Cross-platform software tool for the GUI test automation specialized on Java/Swing, SWT, Eclipse plug-ins and RCP applications, Java applets, Java Web Start, ULC and cross-browser test automation of static and dynamic web-based applications	Proprietary
Ranorex	http://www.ranorex.com/	GUI test automation framework for testing of desktop, web-based and	Proprietary

		mobile applications based on record&playback.	
Rational Functional Tester	http://www-03.ibm.com/software/products/en/functional	IBM Rational Functional Tester is an automated functional testing and regression testing tool. This software provides automated testing capabilities for functional, regression, GUI, and data-driven testing	Proprietary
Robot Framework	http://www.robotframework.org/	The Robot Framework is a generic test automation framework for acceptance testing and acceptance test-driven development. It is a keyword-driven testing framework that uses tabular test data syntax.	OSS (Apache 2.0 license)
Sahi	http://sahipro.com/	Sahi is an automation and testing tool for web applications coming in an open-source and a proprietary version. The open-source version includes a basic tool set sufficient for most testing purposes (Record on all browsers, Playback on all browsers, HTML playback reports, JUnit Style playback reports, Suites and batch run, Parallel playback of tests), whereas the Pro version includes further features such as test distribution and report customization.	OSS/Propietary
Silk	http://www.borland.com/products/silktest/	SilkTest is a tool for automated function and regression testing of enterprise applications. It offers various clients: SilkTest Classic, Silk4J (Java), Silk4Net (VB or C#), SilkTest Workbench (VB.Net).	Proprietary
SOAtest	http://www.parasoft.com/api-testing	Tool suite for testing and validating APIs and API-driven applications (e.g., cloud, mobile apps, SOA).	Proprietary
Squish	https://www.froglogic.com/squish/	Cross-platform GUI and regression testing tool that can test applications based on a variety of GUI technologies. It is developed and maintained by Froglogic.	Proprietary
Test Studio	http://www.telerik.com/teststudio	Telerik Test Studio is a Windows-based software testing tool for web and desktop functional testing, software performance testing, load	Proprietary

		testing and mobile application testing developed by Telerik.	
TestComplete	https://smartbear.com/product/testcomplete/	TestComplete is a functional automated testing platform to create automated tests for Microsoft Windows, Web, Android, and iOS applications. Tests can be recorded, scripted or manually created with keyword driven operations and used for automated playback and error logging.	Proprietary
Unified Functional Testing (UFT)	https://saas.hpe.com/en-us/software/uft	HPE Unified Functional Testing (UFT) software, formerly known as HP QuickTest Professional (QTP), provides functional and regression test automation for software applications and environments. HPE Unified Functional Testing supports keyword and scripting interfaces and features a graphical user interface.	Proprietary
Watir	http://watir.github.io/	It is an open source Ruby library for automating tests. Watir interacts with a browser the same way people do: clicking links, filling out forms and validating text.	OSS (MIT)
Xnee	https://xnee.wordpress.com/	It is an open source tool to record, replay and distribute user actions under the X11 environment.	OSS (GPL)
qTest Explorer	https://www.qasymphony.com/software-testing-tools/qtest-explorer/test-execution-recorder/tab/record/	Tool for Test Faster, Report Bugs, and Automate Test Execution on desktop, web or mobile applications.	Proprietary

Table 19. GUI automation tools

4.5.1.1 *Selenium*

It is the most popular framework able to perform automated web application testing cross-browser and cross-platform. The main features of Selenium are: i) it is open source; ii) it has a client/server architecture; iii) it provides a domain-specific language (DSL) for tests; iv) it provides Selenium WebDriver remote control interface; v) it

supports many programming languages and browsers (the most of high-level applications are based on it).
Specifically, it supports the following browsers (there are many WebDriver Implementations): ChromeDriver, EventFiringWebDriver, FirefoxDriver, HtmlUnitDriver, InternetExplorerDriver, PhantomJSDriver, RemoteWebDriver, SafariDriver.
Supported programming languages are: Java, C#, Python, Ruby, PHP, Perl and JavaScript. It run in main operating system (Windows, Linux and Apple OS X) and provides Selenium Grid that enables running tests on many servers at the same time, cutting down on the time it takes to test multiple browsers or operating systems. It provides Selenium IDE that is an extension of Firefox and allows to record, edit and debug tests. Moreover, Selenium WebDriver is going to become a standard by W3C[57] .

4.5.1.2 *Appium*

Appium is one of the most popular cross-platform (Android and iOS) test automation framework for automated testing of native and hybrid mobile applications. It is considered the Selenium for mobile. The main features of Appium are: Open Source; Client/Server Architecture; Use WebDriver API as interface for Appium clients; it allows to write test in many languages (Java, C#, Haskell, JavaScript, Objective-C, Perl, Perl, Python, R and Ruby); it runs in main operating system (Windows, Linux and Apple OS X); many high-level mobile testing automation tools are based on Appium.

4.5.1.3 *Nightwatch.js*

It is a framework based on Node.js and Selenium WebDriver, to testing web applications from JavaScript clients. The main features are: i) it is open Source; ii) it uses Selenium WebDriver API for Browser interactions; iii) it builds tests using JavaScript, CSS or XPath selectors; iv) Grunt Support; v) Cloud Service Support; vi) it works with Sauce Labs and BrowserStack; vii) Continuous Integration Support. Provide JUnit XML reporting for test execution results; viii) it is easy to extend adding new commands and assertions.

4.5.1.4 *qTest Explorer*

qTest Explorer is a proprietary module of qTest Cloud Platform, to test faster, report bugs and automate test execution. Among its main features there are: i) it is able to test Desktop, Web or Mobile application; ii) enabling to document test cases while you are testing; iii) sharing results with popular agile ALMs like JIRA, VersionOne, Rally and more; iv) enabling record test sessions in desktop, web and mobile application; v) converting a test session, into an automated Selenium (java, C# or Python) or Protractor script; vi) providing integration with the central repository for the automatized test scripts; vii) integration as plugin with both Firefox and Chrome.

4.5.1.5 *TestComplete*

It is an automated testing platform, part of Smarbear's suite. The main features of TestComplete are: i) Test on Desktop, Web or Mobile; ii) Windows Support; iii) Support for multiple scripting languages; iv) Keyword Driven Testing; v) Automated test

[57] https://www.w3.org/TR/webdriver/

recording; vi) Automated Test Visualizer. TestComplete documents each operation performed during test execution with screenshots; vii) Extendable by plugins; viii) Provides Test Editor and Debugger; ix) Object-Based GUI Recording; x) Test Report; xi) Integrated Unit Testing for wide variety of languages for your desktop, web or mobile applications; support for Data-Driven Testing and Regression Testing; xii) Integration with LoadComplete (another module of the platform) to convert functional test to performance test; xiii) Integration with CrossBrowserTesting, to run and report across thousands of test environments at once. It allows to combine automated tests with manual and visual tests in the cloud lab to improve coverage.

Table 20 compares the main GUI automation tools and frameworks. From this table, we can see that there are tools to make automated tests very diverse. Some of them with language of scripting such as java, JavaScript, python and so on, some for desktop applications, other for web applications and others for mobile applications.

	Selenium	Appium	Nightwatch.js	qTest	TestComplete
Test Creation Tool	No	No	No	qTest Explore	TestComplete IDE
Type of application to test	Web	Mobile	Web	Desktop Mobile Web	Desktop Mobile Web
Supported System (testing system)	Windows, Linux and OS X.	Windows, Linux and OS X	Windows, Linux and OS X	Windows	Windows
Supported System (tested system)	Web (cross-browser)	Android, iOS	Web (cross-browser)	Web (cross-browser), Windows	Web (cross-browser), Windows, Android, IOS
Scripting language	Java, C#, Python, Ruby, PHP, Perl and Javascript	Java, C#, Python, Ruby, PHP, Perl and Javascript	Javascript	Java, C# , Python, Protractor	JScript Python VBScript DelphiScript C# and C++Script
Codeless testing tool	No	No	No	Yes	Yes

Integrations	The most of high level tools for Web App Testing, use Selenium	Selenium	Selenium	qTest Management ,JIRA, VersionOne, Rally and more	CrossBrowserTesting
				Advantage	*Disadvantage*

Table 20. Comparison of GUI automation tools

4.5.2 Device emulation - Baseline and comparative analysis

Device emulation refers to the approach where a virtual device can take the role of a real device. The process of emulation can help in establishing a device to completely replace the functionalities of a real device. It depends on the use case, for which the user decides the level of emulation required. The level of emulation here refers to the extent the virtual device mimics the real device in terms of its overall aspects. Table 21 shows a list of services and tools that can assist or provide emulated IoT devices in order to realize and test IoT applications. Devices in the context of IoT, refer to sensor, actuators and smart devices. The area of IoT testing with device emulation is interesting because, it reduces the cost to test IoT solutions that can be complex and may require many devices. In order to sufficiently address the number of devices required to realize a complex application, device emulation helps in providing a solution to first test concepts with virtual devices. The view point considered here is that, the sensors or actuators cannot be used directly to realize an IoT application. Associated to the sensors and actuators, a nodal device is required which acts as an interface for the network. Sensor and actuator data is obtained by the nodal device that converts them to for example data packets or visualized on a GUI.

Name	URL	Short Description	License
Contiki	http://www.contiki-os.org/	An OS and a platform used to program microcontrollers used in wireless sensor nodes.	OSS (3-Clause BSD)
Contiki Cooja Simulator	https://github.com/contiki-os/contiki/wiki/An-Introduction-to-Cooja	An open source simulator for wireless sensor networks. The platform provides emulated nodes comprising microcontrollers and associated sensor hardware. Standard sensor nodes such as TelosB are provided as a whole on which it is possible to run binaries.	OSS (3-Clause BSD)
Iotify	https://iotfy.io/	A cloud platform, which emulates edge platforms such as Raspberry Pi	Private

		and associated sensor devices. In a virtual lab, it is possible to attach the required device to the edge platform to realize an edge node. The edge device can be programmed using applications which can access the hardware interface via the OS of the edge platform. The data is sent to and can be visualized on the cloud.	
ns3	https://nsnam.org/	ns-3 is a network simulator. It can be used to simulate network conditions between virtual nodes. Although the main focus is on network simulation, ns-3 could be used to emulate IoT devices by programming the nodes.	OSS (GPL-2.0)
AWS Lambda + AWS IoT	https://aws.amazon.com/blogs/iot/device-simulation-with-aws-iot-and-aws-lambda/	This is a combination of services available from AWS. AWS is SaaS which is used to visualize IoT data. IoT devices can be emulated on top of AWS Lambda and together with AWS Lambda and AWS IoT, it is possible to realize an IoT device emulation.	Private
Contiki mspsim	https://github.com/contiki-ng/mspsim	This complements Contiki Cooja simulator described earlier. mspsim offers customization to standard MSP430 microcontrollers which can be used to customize emulated on-board hardware devices for example on TelosB.	OSS (Copyright 2007, Swedish Institute of Computer Science.)
Azure device simulation	https://github.com/Azure/device-simulation-dotnet	It is a container based .NET implemented, it supports device to cloud and cloud to device message exchange platform built on top of Azure IoT Hub. The device model which states the frequency and data format is explicitly written to provide a specific virtual device.	MIT
Ixia IoT Testing	https://www.ixiacom.com/products/ixia-iot-iot-device-testing	Focusing on the network between nodes, this tool focuses on emulating the network conditions.	Private
NetSim	https://www.tetcos.com/netsim-iot.html	It focuses on emulating the network between source and destination nodes, with actual sensor nodes at the source side and data analyzed at	Private

		the destination node. The NetSim resides between the source and destination nodes.	
Azure IoT Edge	https://docs.microsoft.com/en-us/azure/iot-edge/	Container based method to emulated IoT devices on the edge node instead of the cloud. This complements the Azure device simulation by analyzing the data on the edge node instead of the cloud.	Private
TinyOS	https://github.com/tinyos/tinyos-main	It is an OS used to write wireless sensor networks applications on top of standard sensor nodes such as TelosB.	OSS (BSD)
OpenMTC	https://github.com/OpenMTC/OpenMTC	OpenMTC is an implementation of the machine to machine communication standard.	OSS (EPL-1.0)

Table 21. Device emulation tools

Among the tools presented in Table 21, the most relevant ones are: Contiki Cooja Simulator, iotify, ns-3, AWS (IoT + Lambda), Contiki mspsim, Azure device simulation, Azure IoT Edge, OpenMTC. We present below a description of such tools mainly considering how they emulate the behaviour of the nodal devices. Finally, Table 22 shows a comparison among them. In particular, these tools are compared according to the following aspects:

Prerequisite: Any application that the user needs to run to execute and access the facilities to get started with device emulation.

Language: Language/s in which the application is written.

Codeless testing tool: Whether there exists a codeless testing tool such as buttons or another graphical interface which could be used to turn on/off or change behavior of the device.

Type of device emulated: It refers to the type of device emulated. Type of device refers to sensor/actuator hardware or platform.

Runtime environment emulated device: It refers to where the application used to emulate the device is run.

Runtime environment IoT application: Where the IoT application which makes use of the emulated devices is run ultimately.

Interface: The type of interface used for communication between the IoT application and the emulated devices.

4.5.2.1 *Contiki Cooja Simulator*

This is a graphical simulator[58]. A user can use the compiled binary for the sensor node typically used in Wireless Sensor Networks (WSN). These nodes are made up of microcontrollers interfacing with sensors. Its main features are: I) a GUI is used as the only source of interaction; II) It runs in main operating system (Linux, OS X, Windows); iii) Once the sensor node type is selected and the node is programmed with the binary, the user can request for any number of such nodes and start the simulation. When simulation is in progress, the user can monitor various parameters of the node, such as battery level, radio frequency signal strength of each node collectively or separately; iv) it provides required number of nodes of a particular platform. Ex. For TelosB, these emulated sensor nodes are available as part of Cooja; v) the user is able to modify the behavior of the provided features in the nodes.

4.5.2.2 *Iotify*

Iotify[59] is a cloud-based platform for emulation of hardware devices used in realizing IoT applications. Iotify offers a so-called platform called Virtual Lab where a user can choose a preferred hardware device to construct a suitable node. It is a closed and cloud-based platform and uses a GUI based programming environment. The user can program the selected node (e.g. Raspberry Pi), to interact with the underlying hardware devices. The node can be replicated and connected in the cloud infrastructure. The method of interaction with the user once the application is deployed in the cloud is with REST API. Emulation aspect is that, the hardware platform (e.g. Raspberry Pi) can be chosen and programmed to interact with a preferred sensor available also as a hardware device. It also allows for data analytics and visualizations facilities.

4.5.2.3 *NS-3*

Popular among the network engineers, ns-3[60] is a tool used to simulate networks. It is possible to program nodes, interconnect the nodes with the preferred medium of networking (Wired or wireless) and simulate the network by varying the network conditions. The method of interaction is dependent on the user (can use REST or standard output) and a binary is run on the user machine to start the simulation. It runs in main operating system (Linux, OS X, Windows). Method of emulation is that the user can program the individual nodes to the advantage of realizing an IoT application. Each node needs to be programmed by the user.

4.5.2.4 *AWS IoT + AWS Lambda*

AWS Lambda[61] can be used for realizing the function of an emulated device and for interconnecting the devices built using AWS Lambda. It is closed and cloud based and it is able to interact with standard AWS services. As described before, the user is able to

[58] https://github.com/contiki-os/contiki/wiki/An-Introduction-to-Cooja

[59] https://iotfy.io/

[60] https://nsnam.org/

[61] https://aws.amazon.com/blogs/iot/device-simulation-with-aws-iot-and-aws-lambda/

interconnect emulated AWS Lambda devices with AWS IoT and needs to write the features required for each type of emulated device. It allows for data analytics and visualization services as a result of AWS IoT.

4.5.2.5 *Contiki MSP Sim*

This approach[62] is similar to the already described tool, Contiki Cooja Simulator and includes all the aspects related to simulation. The main features are: i) GUI is used as the only source of interaction; ii) it runs in main operating system (Linux, OS X, Windows) and the provided hardware nodes can be selected by the user; iii) this tool enables to change the features provided by Cooja MSP 430 microcontroller series; iv) Emulation is that the user can change the features of the nodes provided in Cooja by changing the instructions provided to MSP 430 microntrollers. For instance, the user can add an extra sensor or remove one using this tool. Such modified unit can be used in simulation.

4.5.2.6 *Azure device simulation*

Azure device simulation[63] works with Azure IoT Hub. It is closed and cloud based. It interacts by using standard Azure services. Azure device simulation is used to emulate behavior of a node or a collection of nodes by using a user defined program deployed on the azure cloud. The nodes deployed as explained before relay telemetry data to Azure IoT hub. The user can realize IoT applications using the features of IoT hub. Emulation refers to the facility offered by Azure to program and deploy nodes in the cloud. It provides Data analytics and visualizations as part of Azure IoT Hub.

4.5.2.7 *Azure IoT Edge*

This service[64] works with Azure IoT Hub. Similar to Azure device simulation, except that the nodes are actual edge devices of the user instead of cloud. It is closed and cloud based. The method of interaction is by using standard Azure services. Additionally, interaction is enabled by using REST API. The device emulation can be realized by deploying docker containers on user's edge devices. The edge devices send data to Azure IoT hub where the IoT application can be realized. Emulation refers to the facility offered by Azure to program the edge devices using docker containers. The user can program the edge device to emulate the desired behavior from the edge device. The user is also required to program the image. Data analytics and visualizations are provided as part of Azure IoT hub.

4.5.2.8 *OpenMTC*

OpenMTC[65] is a middleware which can be used to program and interconnect nodes. In particular, nodes can be individually programmed to emulate device behavior. The interaction happens using REST API. It runs in main operating system (Linux, OS X,

[62] https://github.com/contiki-ng/mspsim

[63] https://github.com/Azure/device-simulation-dotnet

[64] https://docs.microsoft.com/en-us/azure/iot-edge

[65] https://github.com/OpenMTC/OpenMTC

Windows) and provides a middleware framework using the oneM2M industry communication standard. Using python, then user can implement each node with emulated behavior devices. Those programmed nodes can be interconnected to realize an IoT application. Emulation refers to the capability offered by OpenMTC to implement custom device behavior on nodes using children programs embedded in the program that is used to connect the node to the communication network.

	Contiki Cooja Simulator	iotify	ns-3	AWS (IoT + Lambda)	Contiki MSPSim	Azure device simulation	Azure IoT Edge	OpenMTC
Prerequisite	Java, Cooja installation	Virtual IoT lab and associated tools from iotfy	ns-3 installation	AWS resources	Java, Cooja installation	Azure device simulation installation, Azure IoT Hub subscription	Docker, Azure IoT Hub subscription	Python, OpenMTC installation
Language	nesC, C	Python, C, C++	Python, C++	Java Script	Java, C, nesC	.NET	Java, .NET Core 2.0, Node.js, C, and Python	Python
Codeless testing tool	Yes	NA	No	No	No	NA	NA	No
Type of emulated device	Factory defined. Hardware platform such as sensor node. Eg. TelosB sensor node module	Factory defined. Generic hardware platform. Eg. Raspberry Pi.	User defined. User can implement device nodes to mimic nodal behavior at the network level.	User defined. User can implement device nodes to mimic the nodal behavior at the network level.	User defined. Able to modify the platform at the hardware instruction level. Hardware platform such as sensor	User defined. User can implement device nodes to mimic nodal behavior at the network level	User defined. User can implement device nodes to mimic nodal behavior at the network level.	User defined. User can implement device nodes to mimic nodal behavior at the network level.

				node. Eg. TelosB sensor node module			
Runtime environment for emulated device	Linux, OS X, Windows	Cloud	Linux, OS X, Windows	Cloud	Linux, OS X, Windows	Docker Container. Linux, OS X, Windows	Docker Container. Linux, OS X, Windows	Linux, OS X, Windows
Runtime environment for IoT application	Linux, OS X, Windows	Cloud	Linux, OX X, Windows	Cloud	Linux, OS X, Windows	Cloud	Cloud	Linux, OS X, Windows
Interface	GUI	REST API	Restricted to application. Can be extended if user wishes so.		GUI	REST API, SDK	REST API, SDK	REST API, SDK

Table 22. Comparison of device emulation tools

4.5.3 Progress within ElasTest

Progress in GUI automation and impersonation in based on two principles: (1) the creation of an advanced FOSS user impersonation as a service capability that provides SaaS GUI automation basing on open source and standard paradigms. (2) the creation of an equivalent automation capability for sensors, actuators and smart devices that enables emulating the behavior of such components on all types of IoT SUTs. A prototype is already available in the current version.

4.6 Cloud Instrumentation

4.6.1 Baseline and comparative analysis

Cloud Instrumentation refers to the management and orchestration of services, in particular, virtualized resources within the cloud. In recent years, this topic became very important in research and industry due to the ongoing increase of cloud-based services in order to simplify the management and orchestration of those network services. Many solutions emerged over the last time which is offering the virtualization of resources whereas the concepts, flexibility and also the management of such tools varies. Various features of those solutions must be considered in order to find a proper candidate which satisfies the ElasTest requirements.
Table 23 shows a list of cloud instrumentation tools. Among them, the most relevant ones are: OpenStack, Amazon Web Services (AWS), Docker, LXD, OpenShift, Kubernetes, Open Baton, Juju.
Table 24 shows a comparison of these cloud instrumentation tools. The following characteristics will be used for proper comparison of the proposed solutions:

1. *Model of abstraction*: Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS), Container-as-a-Service (CaaS);
2. *Type of virtualized resources*: Virtual Machines, Containers;
3. *Monitoring support*: internal, external;
4. *Runtime Management*: autoscaling of resources, update of services;
5. *Activity;*
6. *License*: OSS, Proprietary;
7. *Management*: API, CLI, SDK, Dashboard.

In the following, a short description of these tools is presented.

Name	URL	Short Description	License
Deltacloud	https://deltacloud.apache.org/	It is an API developed by Red Hat and the Apache Software Foundation that abstracts differences between cloud computing implementations	Apache

Apache juju	https://jujucharms.com/	It is an application and service modelling tool that enables you to quickly model, configure, deploy and manage applications in the cloud with only a few commands	GNU Affero GPL v3
Openstack	https://www.openstack.org/	OpenStack is an open source project which offers the virtualisation of resources (compute, storage, network) by making use of a very modularized architecture.	OSS (Apache 2.0)
Cloudstack	https://cloudstack.apache.org/	Apache CloudStack is open source software designed to deploy and manage large networks of virtual machines, as a highly available, highly scalable Infrastructure as a Service (IaaS) cloud computing platform.	OSS (Apache 2.0)
Docker	https://www.docker.com/	Docker enables the execution of applications as a lightweight container.	OSS (Apache 2.0)
Vagrant	https://www.vagrantup.com/	Open-source software product for building and maintaining portable virtual development environments.	OSS (MIT)
Kubernetes	https://kubernetes.io/	Kubernetes is an open-source system for automating deployment, scaling, and management of containerized applications.	OSS (Apache 2.0)
Amazon Web Services (AWS)	https://aws.amazon.com/	Amazon Web Services (AWS) provides on-demand computing resources and services in the cloud, with pay-as-you-go pricing.	ASL (Amazon Software License)
OpenNebula	https://opennebula.org/	OpenNebula is an open-source project delivering a simple but feature-rich and flexible solution to build and manage enterprise clouds and virtualized data centers.	OSS (Apache 2.0)
CloudStack	https://cloudstack.apache.org/	Apache CloudStack is open source software designed to deploy and manage large networks of virtual machines, as a highly available, highly scalable Infrastructure as a	OSS (Apache 2.0)

		Service (IaaS) cloud computing platform.	
OPNFV	https://www.opnfv.org/	Open Platform for NFV (OPNFV) facilitates the development and evolution of NFV components across various open source ecosystems. Through system level integration, deployment and testing, OPNFV creates a reference NFV platform to accelerate the transformation of enterprise and service provider networks.	OSS (Apache 2.0)
openvim	https://osm.etsi.org/	OpenVIM is a lightweight implementation of an NFV VIM supporting EPA (Enhanced Platform Awareness) features and control of an underlay switching infrastructure.	OSS (Apache 2.0)
OVX	http://ovx.onlab.us/	OVX is a network hypervisor that can create multiple virtual and programmable networks on top of a single physical infrastructure. Each tenant can use the full addressing space, specify their own topology, and deploy the network OS of their choice. Networks are reconfigurable at run-time, and OVX can automatically recover from physical network failures.	OSS (Apache 2.0)
Cloudify	http://getcloudify.org	Cloudify is an enterprise-class open source Platform as a Service (PaaS) stack providing the full end-to-end lifecycle of NFV orchestration through a simple TOSCAbased YAML blueprint following a topology-driven and application-centric approach.	OSS (Apache 2.0)
LXD	https://linuxcontainers.org/lxd/	LXD is a container-based hypervisor.	OSS (Apache 2.0)
Docker	https://www.docker.com/	Docker enables the execution of applications as a lighweight container.	OSS (Apache 2.0)

OpenShift	https://www.openshift.com/	OpenShift is a distribution of Kubernetes optimized for application development and multi-tenant deployment.	OSS (Apache 2.0)
Juju	https://jujucharms.com/	Juju is an open source project based on a universal service modeling system with a service-oriented architecture and service-oriented deployment. It allows to deploy, congure, manage, maintain and scale cloud services easily and efficiently on public clouds, physical servers, OpenStack and containers.	AGPL v3
OpenStack Heat	https://docs.openstack.org/developer/heat/	Heat is the main project in the OpenStack Orchestration program. It implements an orchestration engine to launch multiple composite cloud applications based on templates in the form of text files that can be treated like code.	OSS (Apache 2.0)
OpenStack Tacker	https://docs.openstack.org/developer/tacker/	Tacker is an official OpenStack project building a Generic VNF Manager (VNFM) and a NFV Orchestrator (NFVO) to deploy and operate Network Services and Virtual Network Functions (VNFs) on an NFV infrastructure platform like OpenStack. It is based on ETSI MANO Architectural Framework and provides a functional stack to Orchestrate Network Services end-to-end using VNFs.	OSS (Apache 2.0)
CORD	http://opencord.org/	CORD (Central Office Re-architected as a Datacenter) combines NFV, SDN, and the elasticity of commodity clouds to bring datacenter economics and cloud agility to the Telco Central Office. CORD lets the operator manage their Central Offices using declarative modeling languages for agile, real-time configuration of new customer services. Major service providers like AT&T, SK	OSS (Apache 2.0)

		Telecom, Verizon, China Unicom and NTT Communications are already supporting CORD.	
Kubernetes	https://kubernetes.io/	Kubernetes is an open-source system for automating deployment, scaling, and management of containerized applications.	OSS (Apache 2.0)
RIFT.ware	https://open.riftio.com/	RIFT.ware the next generation Network Service Virtualization (NSV) platform that delivers management and orchestration (MANO) and automation of virtual network services, applications, and functions at scale. RIFT.ware is a model-driven, ETSI compliant NFV MANO solution that simplifies deployment of VNFs and the composition and management of complex network services.	OSS (Apache 2.0)
Open Baton	http://openbaton.github.io/	Open Baton is an ETSI Network Function Virtualization (NFV) compliant Framework offering high flexibility and modularity for the Management and Orchestration (MANO) of virtualized Network Functions (VNFs).	OSS (Apache 2.0)
Open Source MANO (OSM)	https://osm.etsi.org/	OSM is providing an open source Management and Orchestration (MANO) stack following the ETSI NFV Information Models.	OSS (Apache 2.0)
Open-O	https://www.open-o.org/	OPEN-Orchestrator Project (OPEN-O) framework and orchestrator enables agile software-defined networking (SDN) and network function virtualization (NFV) operations.	OSS (Apache 2.0)
Sonata	http://sonata-nfv.eu/	SONATA is developing a NFV framework that provides a programming model and development toolchain for virtualized services, fully integrated with a DevOps-enabled service platform and orchestration system.	OSS (Apache 2.0)

Gohan	http://gohan.cloudwan.io/	Gohan makes REST-based API server, database backend, and WebUI from JSON for a simple and unified cloud service architecture having NFV MANO as a Use Case.	OSS (Apache 2.0)
ARIA	http://ariatosca.org/	It is an open, light, CLI-driven library of orchestration tools that other open projects can consume to easily build TOSCA-based orchestration solutions.	OSS (Apache 2.0)
VirtualBox	https://www.virtualbox.org/	It is a powerful hypervisor, easy to set up and use on a variety of operating systems, provides GUI, CLI and API management options.	GNU GPL v2
KVM	http://www.linux-kvm.org/	It is an hypervisor built into the Linux kernel, makes use of hardware-assisted virtualization, integrates with QEMU for better management support and can host Linux and Windows images.	GNU GPL v2

Table 23. Cloud Instrumentation tools

4.6.1.1 *OpenStack*

OpenStack[66] is an open source project providing a rich set of functions to control large pools of compute, storage and networking resources. It follows a very modular architecture where each module offers a certain set of functionalities as shortly described below:

- Keystone: Identity Management
- Nova: Management of compute resources
- Neutron: Management of networking resources
- Glance: Management of images
- Horizon: OpenStack Dashboard
- Swift: Management of block storage
- Cinder: Management of object storage

***Model of abstraction**:* OpenStack can be seen as an Infrastructure-as-a-Service since it provides all the capabilities to deploy and manage virtual machines on top of a physical resource. Besides the basic OpenStack functionalities, there are also other original orchestration solutions which are closely bound to the main OpenStack components, such as, Heat and Tacker.

[66] https://www.openstack.org/

Type of virtualized resources: The type of virtualized resources is dependent on the Hypervisor in use. Initially, OpenStack supported KVM but step-by-step the set of supported Hypervisors was statically extended by using some kind of compute driver. Nowadays, even container deployments are supported, such as, LXD or Docker or other compute drivers[67].

Monitoring support*: B*y default, OpenStack includes a default monitoring module, namely Ceilometer. Ceilometer offers predefined meters which can be monitored. These meters are basically retrieved from other modules such as the compute module (Nova) which provides computer-related monitoring information, for example, usage of CPU, memory and disk space. Additionally, the administrators can define new meters to monitor as well.

Runtime Management*: C*lassical runtime management in terms of automated management and orchestrations is not provided by the main OpenStack modules, except Heat and Tacker which are covered. Nevertheless, OpenStack can be requested manually to execute basic operations on running instances such as pause, suspend, shut off and resume, resize, and attach interfaces.

Activity*:* OpenStack was announced in 2010 as a joint project between the Nasa and Rackspace Hosting. Nowadays, OpenStack is managed by the OpenStack Foundation and become one of the most used software platforms with regard to virtualization solutions. Hence, the community around OpenStack became rather large with a lot of activities in different directions. Regular events like the OpenStack summits gained a lot of worldwide attraction.

License*:* OpenStack is completely open source (hosted on GitHub) and released under the Apache License 2.0.

Management*:* OpenStack provides several management options. As mentioned before, Horizon is the web-based user interface. Additionally, OpenStack exposes a well-defined API and provides directly a python-based client. Besides the official management tools, there are a lot of third-party toolkits and libraries written in several languages (e.g. Java, Go, Erlang, C, C++, etc.).

4.6.1.2 *Amazon Web Services (AWS)*

Amazon Web Services (AWS)[68] is the commercial virtualization product offered by Amazon. It provides more than 70 infrastructure services for addressing a broad set of functions. Besides the most common services for compute, storage and networking resources, it offers also database, analytics, deployment, management, mobile and applications services, just to name the most common ones. Furthermore, Amazon provides a worldwide distributed server farm, so called regions, which allows the customer to deploy services in chosen datacenters. For charging, Amazon follows the pay-as-you-go principle which charges only the amount of resources which are really used.

Model of abstraction: Due to the large set of services, it makes it rather impossible to put Amazon into any kind of cloud model. Actually, it offers all kinds of cloud models

[67] https://docs.openstack.org/developer/nova/support-matrix.html

[68] https://aws.amazon.com/

which depend on the chosen service. Infrastructure-as-a-Service can be achieved by using the EC2 module for managing virtual instances only; whereas Platform-as-a-Service and Software-as-a-Service can be achieved by selecting the appropriate service offered by AWS. For instance, using and combine exposed services for hosting certain web services or services for mobile devices within a single click.

***Type of virtualized resources**:* AWS offers several compute products which allows the customer to deploy, run and scale deployment units - either virtual servers, containers or even code directly. Depending on the use case, the following services are provided:

- Amazon Lightsail: Host simple applications and websites within a single or few virtual private servers.
- Amazon EC2: Host any application which are running on a set of servers (cluster).
- Amazon ECS: Container-based deployments of stateless or stateful applications defined by Docker containers.
- AWS Lambda: Host event-initiated, stateless applications.

Monitoring support: CloudWatch is the monitoring service which is offered by Amazon. It allows to monitor AWS cloud resources and the applications. It collects and tracks metrics and log files and allows also to set alarms which can be used to react automatically to changes of the AWS resources. Additionally, it provides system-wide observation of resource utilization, application performance and operational health.

***Runtime Management**:* Runtime management of AWS resources can be achieved in several ways. A so-called Auto Scaling Groups can be defined in order to scale a certain group of instances by defining thresholds, corrective actions, cooldowns, termination policies and additional behaviors. Another interesting service is the AWS OpsWorks which enables the runtime management by using Chef as the management tool.

***Activity**:* Since AWS is a proprietary software it is rather difficult to track the development and maintenance activities. Nevertheless, AWS is well-maintained by Amazon with static improvements and introduction of (new) services. As already mentioned, AWS provides more than 70 commercial services which is a strong indicator with respect to the activity.

***License**:* AWS is a proprietary software making use of the Amazon Software License (ASL).

***Management**:* Besides the web-based management and command line tool, Amazon provides a lot of self-maintained libraries and SDKs for different programming languages, for instance, Java, Python, Ruby, Go, C++, and many more.

4.6.1.3 *Docker*

Docker[69] is the world's leading software container platform. Containers do not bundle a full operating system, but instead only libraries and settings required to make the software work. This makes containers efficient, self-contained systems that guarantee the software will always work the same, regardless of where it is deployed.

Docker can be used for different purposes depending on the point of view from the user:

[69] https://www.docker.com/

- Developers:
  - use Docker to eliminate “works on my machine” problems when collaborating on code with co-workers.
  - Shipping code for building and debugging made simpler
  - Makes setting up and configuring development environments faster and easier
- Operators
  - use Docker to run and manage apps side-by-side in isolated containers to get better compute density.
  - IT ops teams using Docker benefit from faster shipping, testing and deployment
  - guarantee the images work the same in development, staging and production
- Enterprises
  - use Docker to build agile software delivery pipelines to ship new features faster, more securely and with confidence for both Linux and Windows Server apps.

Model of abstraction: Docker provides an Operating-system-level virtualization. This is a virtualization method in which the kernel of an operating system allows the existence of multiple isolated user-space instances. These instances are often called containers.
Type of virtualized resources*:* Containers are an abstraction at the app layer that packages code and dependencies together. Multiple containers can run on the same machine and share the OS kernel with other containers, each running as isolated processes in user space. Containers take up less space than VMs (container images are typically tens of MBs in size) and start almost instantly. Because containers virtualize the operating system and not the hardware (like Virtual Machines), they are more portable and efficient.
Monitoring support: A containers runtime metrics can be streamed in real time using the docker stats command. The information provided contains: CPU usage, Memory usage, Memory limit, Network IO metrics. There are also more than one other solutions through third-party monitoring services that can be used with Docker.
Runtime Management**:** Docker does not provide runtime management in the classical sense.
Activity*:* Docker started as an internal project in dotCloud, a Platform-as-a-Service company and was released as open source in March 2013. On March 13, 2014, with the release of version 0.9, Docker dropped LXC as the default execution environment and replaced it with its own libcontainer library written in the GO programming language. As of 2016 Docker has more than 8 Billion pulls on GitHub.
License**:** Docker has a Community Edition that is licensed under the Apache 2.0 and is hosted on GitHub and a paid Enterprise Edition that provides more support features.
Management**:** Besides the very comprehensive command line tool, Docker provides also an API service and packages for it in python and go. Everything that can be done with the command line tool can also be done via the API, for example: Running and managing containers; Managing Swarm nodes and services; Reading logs and metrics;

Creating and managing Swarms; Pulling and managing images; Managing networks and volumes.

4.6.1.4 *LXD*

LXD [70] can be seen as a lightweight container hypervisor which is based and build on top of LXC. LXC is a userspace interface for the Linux kernel containment features. It is comprised of three main components:

- A system-wide daemon (lxd). It exports a REST API locally and over the network, and provides support for remote managing of container.
- A command line client (lxc). This tool manages all the containers, handles connect to multiple containers hosts and provides a global overview of all the containers defined on the network.
- An OpenStack Nova plugin (nova-compute-lxd), which makes it possible to use LXD hosts as compute nodes which can run workloads on containers rather than virtual machines.

LXD offers the following main features: Secure by design, Scalable, Intuitive, Image based, Live migration.

Model of abstraction: LXD by its nature offers Infrastructure-as-a-Service since it can be seen as a dynamic platform for system containers which are similar to physical or virtual machines. So, it offers compute, storage and networking capabilities like traditional datacenters with classic virtual machine experience including common administrative processes running.

Type of virtualized resources: As mentioned already before, LXD provides virtualization making use of containers. From the user perspective, it can be used like a classical virtual machine including potentially the full software stack, from sshd to syslog.

Monitoring support: LXD does not support any monitoring service out-of-the-box but as usual, monitoring can be achieved by installing and configuring any third-party monitoring tool.

Runtime Management: LXD provides runtime management on the IaaS level which means it can manage and control the container state itself but does not provide any capability to interact with the service which is running on top of the container.

Activity*:* LXD is a very active project with a large community. First release was issued in February 2015. Over the last two years there were around 70 minor/major releases on a regular base (weekly/monthly).

License**:** LXD is an open source project under the Apache 2.0 license. All code is hosted on GitHub directly.

Management**:** LXD provides a RESTful API which allows the management of container and images. In addition, the LXD project offers clients based on Go (lxd. Client) and Python (pylxd) whereas third-party contributors submitted clients for Ruby (Hyperkit), Node.js (node-lxd) and Java (jlxd).

[70] https://linuxcontainers.org/lxd/

4.6.1.5 *OpenShift*

OpenShift[71] is a cloud application development and hosting platform which allows the automation of provisioning, management and scaling for applications. Basically, it is a layered system making use of Docker-formatted container images and Kubernetes concepts. The Docker service provides abstraction for packaging and creating Linux-based, lightweight container images whereas Kubernetes takes care about the cluster management and orchestration of containers on multiple hosts.

Model of abstraction: OpenShift provides a Platform-as-a-Service which allows the developers to quickly develop, host and scale applications.

Type of virtualized resources: As mentioned before, OpenShift uses Docker and Kubernetes to allow automated deployment and management of containers in a private data center.

Monitoring support: OpenShift itself does not provide any built-in monitoring support. However, monitoring can be achieved by using any monitoring tool which must be installed and configured manually within the container. In addition, there are several third-party solutions which are targeting the support of monitoring of the hosts and applications.

Runtime Management: OpenShift provides runtime management in terms of scalability. Scalability can be issued horizontally and vertically. Horizontal scaling is accomplished in such way that OpenShift uses HAProxy as a load balancer in order to distribute the incoming requests among the corresponding containers. Vertical scaling scales the containers itself, so providing either less or more resources to the container itself.

Activity*:* The first release of OpenShift was done in May, 2011. Nowadays, there are 71 releases on a regular base with daily commits (~18000 commits) and high activity.

License: OpenShift is completely open source and licensed under the Apache 2.0 license. The complete code base is hosted on GitHub.

Management*:* The OpenShift container platform provides management through a REST-based API exposed by Kubernetes and OpenShift itself. The OpenShift API provides operations that manage the Kubernetes cluster (security and user management, application deployments, image and source builds, HTTP routing, project management) whereas the Kubernetes API exposes methods for running containerized applications, binding persistent storage, enabling service discovery and managing the infrastructure. All of this is also supported by the CLI provided by OpenShift directly. In addition, several clients are available written in Java, Python and Go.

4.6.1.6 *Kubernetes*

Kubernetes[72] is an open source project for automated deployment and management of containerized applications. Main idea is to group containers into logical units which build the application. Besides this, it claims to provide a stable, portable foundation for developing higher-level automation and customized workflows. Just to name a few features, Kubernetes provides container runtime, container orchestration, container-

[71] https://www.openshift.com/

[72] https://kubernetes.io/

centric infrastructure orchestration, self-healing mechanisms, service discovery and load balancing. The following key goals influenced the design decisions:

- Portability: Potentially, Kubernetes can run everywhere - in public or private clouds, on bare metal infrastructures.
- Flexibility: Integration of own solutions is considered by Kubernetes.
- Extensibility: Any additional capability can be integrated by replacing another component and exposing the same interface as exposed by the built-in functionality.
- Automation: Kubernetes supports both declarative control and imperative control. Declarative control is in charge of system's self-healing and automatic capabilities whereas imperative control allows a higher-level orchestration and automation.

Model of abstraction: Kubernetes provides a cloud-native Platform-as-a-Service which allows easy deployment and management of applications based on containers or clusters of them.
Type of virtualized resources: As already mentioned, Kubernetes is a platform for deploying and managing containers exclusively. Therefore, the type of virtualized resources are containers.
Monitoring support: Monitoring support is provided by a built-in addon support which are running typically in Kubernetes itself. Just to name an option, an option would be to use Heapster to monitor the container cluster.
Runtime Management: Runtime management is supported in several ways. A lifecycle API exposes operations to support the orchestration, in particular, for common types of workloads:

- ReplicaSet: simple fungible and stateless app manager;
- Deployment: orchestrates updates of stateless apps;
- Job: ensures that an expected number of jobs are executed successfully;
- CronJob: time-based Jobs which are executed periodically or at a given point in time;
- DaemonSet: ensures that some/all nodes run a copy of the service (cluster services);
- StatefulSet: a controller ensures ordering of deployment and scaling.

Activity: Since its first release in June 2014, Kubernetes became very popular with a big community. More than 200 releases were done since this time - the last one just recently end of March. More than 1000 contributors produced more than 46000 commits.
License: Kubernetes is completely open source and licensed under the Apache 2.0 license. The full source code is hosted on GitHub.
Management: Besides the RESTful API, the Kubernetes team provides client libraries for Go and Python. The community offers also libraries for Clojure, Go, Java, Node.js, Perl, PHP, Python, Ruby and Scala.

4.6.1.7 *Open Baton*

Open Baton[73] is an ETSI NFV compliant orchestration framework driven by Fraunhofer FOKUS and the Technical University Berlin. As shown in Figure 20, it follows a very modular approach with well-defined interfaces between the components. All functional blocks can communicate over the centralized message bus. This enables an easy extensibility by plugging in new components which may have specific management and orchestration capabilities. Another feature is the plugin approach which allows the support of new virtualization infrastructures or monitoring systems. At the moment Open Baton supports VIM drivers for OpenStack and Docker whereas Zabbix is the main monitoring system in use. Nevertheless, new technologies can be supported easily by implementing certain interfaces whereas all the communication happens seamlessly due to the provided SDKs.

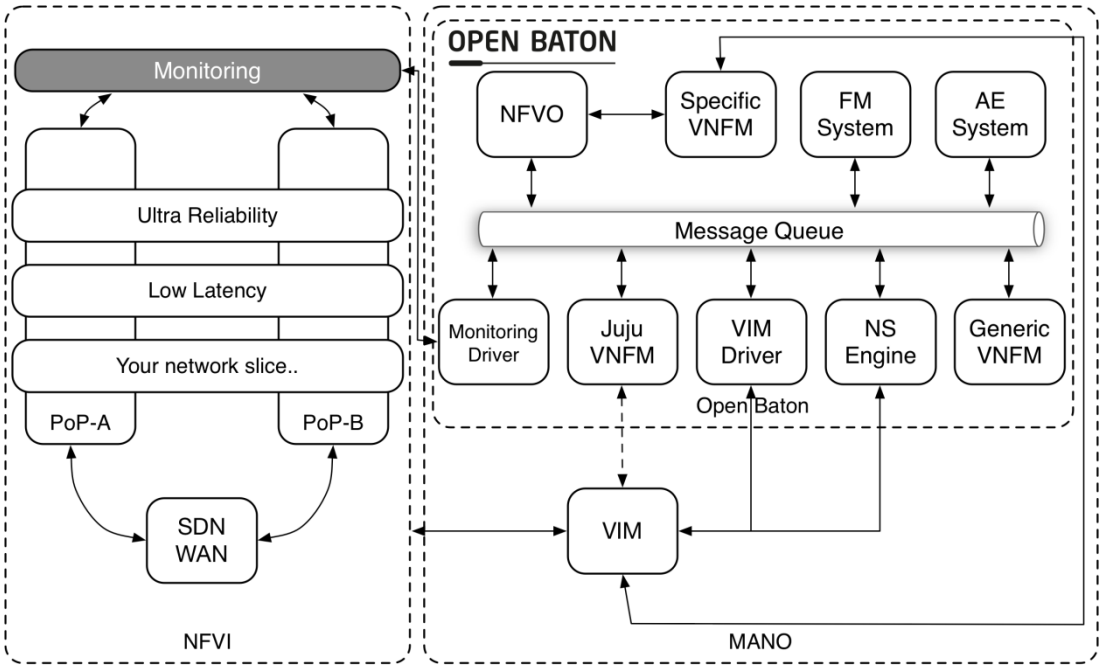

Figure 20. Open Baton Overview

Model of abstraction: Open Baton offers end-to-end service orchestration across different datacenters. Depending on the defined service it may offer IaaS, PaaS and SaaS - all together.

Type of virtualized resources: Based on the virtualization technology underneath, it supports all kinds of virtualized resource - virtual machine and containers. Only requirement is to have a VIM driver available for the targeted infrastructure.

Monitoring support: As already mentioned, Open Baton support at the moment Zabbix as the main Monitoring system where the clients are installed and configured automatically on top of the virtualized resources. Nevertheless, any monitoring system can be supported by providing the correct monitoring driver.

[73] http://openbaton.github.io/

Runtime Management: Runtime Management is provided in several ways. Nowadays, there are external components available which are covering autoscaling, fault management and network slicing. Apart from the existing set of runtime management capabilities, new functional blocks can be integrated easily through the pub/sub mechanism of events and the SDKs provided. Apart from this, Open Baton provides lifecycle management which is achieved by defining actions which are executed at certain points of the lifecycle of a service.
Activity*:* The first version of Open Baton was released in September, 2015. The release plan is supposed to have regular releases every 6 months. Besides major releases, there are also frequent minor releases based on new features added. Daily commits for fixing bugs, improving workflows or adding new features can be tracked on GitHub.
License: Open Baton is completely open source and licensed under the Apache 2.0 license. The full code base is available on GitHub.
Management*:* Open Baton provides several ways for managing the Virtual Network Functions and Network Services. Besides the RESTful API exposed by the Orchestrator, it offers also command line tools written in Java and Python. For developing new components, it can be used the SDK provided which is written in Java or Python.

4.6.1.8 *Juju*

Juju[74] is an open source project driven by Canonical. It is based on a universal service modelling system with a service-oriented architecture/deployment. It allows to deploy, configure, manage, maintain and scale cloud services easily and efficiently on public clouds, physical servers, OpenStack and containers. The key concept of Juju is to make use of so-called Charms which contain all the instructions necessary for installing and configuring cloud-based services. Charms can be composed to a more complex service – also known as Service Composition. Service Composition is one of the most important features of Juju. In particular, it allows the combination of multiple services into a single functional system by providing two mechanisms for such composition: i) Integration of services across network through standard interfaces; ii) Co-locate services on a common machine.
Model of abstraction*:* Mostly, Juju is seen as a Platform-as-a-Service even if it is not a PaaS per definition. However, this is most due to the fact that Juju can be used to run multiple PaaS on top.
Type of virtualized resources: Based on the configuration, Juju can use virtual machines on top of OpenStack, Amazon Web Services, Microsoft Azure or Google Compute Engine, or containers which might be managed by different container virtualization technologies, such as, Kubernetes, LXD, Elastic Stack or etc.
Monitoring support: Monitoring of applications and services can be achieved by integrating any kind of monitoring tool on the service level. There is no built-in tool provided which allows monitoring out of the box.
Runtime Management: Juju provides runtime management within so called hooks. Hooks are executable files which are defined inside the charm itself and can be seen as a script whose execution is triggered at certain points of the charms lifecycle.

[74] https://jujucharms.com/

***Activity**:* Juju is highly active due to the fact that Juju is own of Canonicals solutions for managing and orchestrating services on cloud infrastructures. The release counter reached already more than 100 releases with more than 40000 commits.
License: Juju's source code is open source and hosted on GitHub. However, it is licensed under the AGPL v3 license and the roadmap is not publicly available.
Management: Juju exposes an API which is based on WebSocket. Besides this API, there is a command line interface and graphical user interface which are using these WebSockets in order to give the user the capability to interact with Juju in an easy way. Juju provides also a Python client using an asynchronous approach.

4.6.1.9 *VirtualBox*

VirtualBox[75] is an open-source project developed by Oracle. It provides a cross-platform virtualization solution, so that the user can run Virtual Machines on a number of different Operating Systems (Windows, Mac, Linux and Solaris). It provides a Graphical User Interface, a Command Line Interface called VBoxManage and a python client, for creating and managing Virtual Machines. Some of VirtualBox most relevant features are listed below: i) Extensive hardware support; ii) Creating snapshots of the current state of a Virtual Machine; iii) Managing groups of Virtual Machines; iv) Remote machine display.
Model of abstraction: VirtualBox is a software which provides everything needed to create and run Virtual Machines, therefore it is classified as hypervisor. It integrates well with Vagrant for providing a more portable environment.
***Type of virtualized resources**:* As already mentioned above, Virtualbox enables the user to create and run Virtual Machines.
***Monitoring support**:* VirtualBox provides a built-in monitoring support. It enables the user to collect and retrieve metrics about the running Virtual Machines. Monitoring can be enabled and disabled via the CLI (VBoxManage).
Runtime Management: VirtualBox control the state of the Virtual Machine but does not provide any capability to interact with the service which is running inside the Virtual Machine.
***Activity*:** The VirtualBox base package has been free since 2010 and since then there have been 7 major releases and multiple minor releases.
***License*:** The VirtualBox base package contains the full VirtualBox source code and platform binaries and is licensed under the GNU General Public License, version 2. There is also a VirtualBox Extension Pack, which includes more hardware support and is licensed under VirtualBox Extension Pack Personal Use and Evaluation License, which is a free license for personal, educational or evaluation use, or an Enterprise License, which is a for-fee license that allows most commercial, non-distribution uses restricted by the PUEL.
Management: VirtualBox provides a CLI (VBoxManager) and a Graphical User Interface for creating and starting Virtual Machines. It also provides python and java clients.

[75] https://www.virtualbox.org/

4.6.1.10 *KVM*

KVM (Kernel-based Virtual Machine)[76] is a virtualization software for Linux on x86 hardware consisting of two parts - the KVM kernel module, part of the Linux kernel and the KVM user module. It supports running Linux and Windows images, without the need to modify them and each virtual machine has private virtualized hardware: a network card, disk, graphics adapter, etc. Instead of using CPU emulation, it uses CPU extension via its kernel module, which makes it smaller and easier to set up and use. Other hypervisors such as QEMU integrate with KVM to make use of the kernel module. Some of the main features of KVM include: i) QEMU Machine Protocol - QMP; ii) CPU Hotplug support - Adding CPUs on the fly; iii) PCI Hotplug support - Adding PCI devices on the fly; iv) VMchannel - communication channel between host and guests.

Model of abstraction: KVM is a software which provides everything needed to create and run Virtual Machines, therefore it is classified as hypervisor and works well with other hypervisors such as QEMU to combine the good performance rates of KVM with a more complete management system.

Type of virtualized resources: KVM enables the user to run Virtual Machines on top of a Linux host, but it uses Hardware-assisted Virtualization (HVM). This makes use of hardware capabilities to create a complete isolation layer for the Virtual Machine.

Monitoring support: KVM has no built-in monitoring support but can be used with monitoring platforms such as op5 or the management platform VMManager.

Runtime Management: KVM does not provide runtime management in the classical sense, but when used together with QEMU it can make use of the QEMU Machine Protocol, which is a JSON-based protocol, that allows the user to control the instance.

Activity: KVM was first introduced in 2007 as part of the Linux kernel mainline and since then it has been a part of every new version of the Linux kernel.

***License*:** The KVM kernel module is licensed under the GNU General Public License, version 2. The KVM user module is licensed under the GNU Lesser General Public License, version 2.

***Management*:** The KVM user module is a CLI, which for Ubuntu Lucid (10.04) or later is combined with QEMU under the "qemu-kvm" package. QEMU also provides other options for management like the QEMU Machine Protocol mentioned above.

Table 24 provides a comparison of the main features of the presented cloud instrumentation tools.

[76] http://www.linux-kvm.org/

	Open Stack	AWS	Docker	LXD	Open Shift	Kubernetes	Open Baton	Juju	VirtualBox	KVM
Abstraction	IaaS	Iaas	CaaS	CaaS	PaaS	CaaS	VNFaaS	PaaS	Hypervisor	Hypervisor
Model Virtualized resources	VM, Container	VM, Container	Container	Container	Container	Container	VM, Container	VM, Container	VM	VM
Monitoring support	Ceilometer	Cloud Watch	built-in	(external)	(external)	(addons)	(drivers)	(externalonly)	built-in	(external only)
Runtime Management	Within Tacker, Heat	Yes	No	No	Yes	Yes	Yes	Yes	Yes	Yes
Activity	high	high	high	high	high	high	high	high	high	high
License	OSS (Apache 2.0)	ASL (Amazon)	OSS (Apache 2.0)	OSS (Apache 2.0)	OSS (Apache 2.0)	OSS (Apache 2.0)	OSS (Apache 2.0)	GNU Affero GPL v3	OSS (GNU GPL v2)	OSS (GNU GPL v2)
Management	RESTful API, CLI, GUI, SDKs	CLI, GUI, SDKs	CLI, GUI, SDKs	RESTful API, CLI, GUI, SDKs	RESTful API, CLI, GUI, SDKs	RESTful API, CLI, GUI, SDKs	RESTful API, CLI, GUI, SDKs	WebSocket API, CLI, GUI, SDKs	CLI, GUI, SDKs	CLI, RESTful API

Table 24. Comparison of cloud instrumentation tools

4.6.2 Progress within ElasTest

The ElasTest platform and, in particular, the ElasTest Platform Manager (EPM) abstracts several cloud instrumentation technologies in order to have an intermediate component with a well-defined northbound interface which allows the abstraction of cloud services so that EPM becomes fully agnostic. This means, that the EPM deploys and executes seamlessly cloud services in the target cloud infrastructures where the consumer of the EPM does not need to care about the underlying infrastructure. The EPM makes use of provided capabilities (e.g. autoscaling, healing functionalities, QoS) of certain cloud technologies whereas missing capabilities of those are compensated within the EPM. To make the EPM fully agnostic of the underlying infrastructure and technology in use, the information of deployed services will be abstracted and returned within a common information model.

4.7 Data Ingestion

4.7.1 Baseline and comparative analysis

Data Ingestion is the process of accessing and importing data for immediate use or storage in a database. In this context, Big Data is a blanket term for the non-traditional strategies and technologies needed to gather, organize, process, and gather insights from large datasets. While the problem of working with data that exceeds the computing power or storage of a single computer is not new, the pervasiveness, scale, and value of this type of computing has greatly expanded in recent years.
Depending on the range of different data input to Big Data system and different data ingestion approaches, there is a number of distributed stream processing frameworks that can be used.
Table 25 briefly describes the most relevant data processing tools.

Name	URL	Short Description	License
Hadoop	http://hadoop.apache.org/	The Apache Hadoop software library is a framework that allows for the distributed processing of large data sets across clusters of computers using simple programming models. It is designed to scale up from single servers to thousands of machines, each offering local computation and storage.	Apache License, Version 2.0
Hadoop YARN	https://hadoop.apache.org/docs/current/hadoop-yarn/hadoop-	It is a resource-management platform responsible for managing computing resources in clusters and using them for scheduling of users' applications.	Apache License, Version 2.0

	yarn-site/YARN.html		
Apache Pig	https://pig.apache.org/	It is a platform for analyzing large data sets.	Apache License, Version 2.0
Apache Sqoop	https://sqoop.apache.org/	It is a tool designed for efficiently transferring bulk data between Apache Hadoop and structured datastores such as relational databases.	Apache License, Version 2.0
Hadoop MapReduce	https://hadoop.apache.org/docs/r1.2.1/mapred_tutorial.html	It is an implementation of the MapReduce programming model for large scale data processing.	Apache License, Version 2.0
Hadoop HDFS	https://hadoop.apache.org/docs/r1.2.1/hdfs_design.html	It is a distributed file-system that stores data on commodity machines, providing very high aggregate bandwidth across the cluster.	Apache License, Version 2.0
Apache Spark	https://spark.apache.org/	Spark provides an interface for programming entire clusters with implicit data parallelism and fault-tolerance.	Apache License, Version 2.0
Apache Flink	https://flink.apache.org/	It is a big data processing tool and it is known to process big data quickly with low data latency and high fault tolerance on distributed systems on a large scale.	Apache License, Version 2.0
Apache Samza	https://samza.apache.org/	A framework that helps software developers to handle streams by processing messages as they come in one at a time. The streams get divided into partitions that are an ordered sequence where each has a unique ID. It supports batching and is typically used with Hadoop's YARN and Apache Kafka.	Apache License, Version 2.0
ELK Stack – Elasticsearch	https://www.elastic.co/products/elasticsearch	It is a distributed high-performance search engine.	Apache License, Version 2.0

ELK Stack – Logstash	https://www.elastic.co/products/logstash	It is a tool for log data intake, processing, and output.	Apache License, Version 2.0
ELK Stack-Kibana	https://www.elastic.co/products/kibana	It is a tool to visualize Elasticsearch data and navigate the Elastic Stack.	Apache License, Version 2.0
Cassandra	https://cassandra.apache.org/	It is an open source distributed NoSQL database management system designed to handle large amounts of data across many commodity servers, providing high availability with no single point of failure.	Apache License, Version 2.0
Apache Kafka	https://kafka.apache.org/	It is a distributed messaging system.	Apache License, Version 2.0
Apache Storm	https://storm.apache.org/	It is a real-time, distributed stream processing computation system, supporting machine learning and continuous computation.	Apache License, Version 2.0
Cloud Commander	http://cloudcmd.io	Cloud Commander is a web file manager that allows to manage the server and work with files, directories and programs in browser from any computer, mobile or tablet.	Open Source (MIT License)

Table 25. Data processing tools

Among the ElesTest objectives related to data ingestion there is the need to develop a system to log data of the ElasTest platform from many sources, aggregating it, and writing it to a Hadoop HDFS. The following frameworks could be used: Hadoop, Elastic's open source ELK stack. For distributed stream processing, the following frameworks could be considered: Samza, Spark, Flink, Storm.

About data ingestion on distributed NoSQL database management systems, the following frameworks could be considered: Cassandra, MongoDB, Couchbase, HBase. For visualization and Analytics Tools, the following tools can be considered: Kibana, Grafana, Graphite.

The biggest difference between Apache Storm and Apache Samza comes down to how they stream data to process it.

Apache Storm conducts real-time computation using topology and it gets feed into a cluster where the master node distributes the code among worker nodes that execute

it. In topology, data is passed in between spouts that spit out data streams as immutable sets of key-value pairs.
Apache Samza streams by processing messages as they come in one at a time. The streams get divided into partitions that are an ordered sequence where each has a unique ID. It supports batching and is typically used with Hadoop's YARN and Apache Kafka.
At the moment, the described data ingestion systems do not refer to standard/shared inbound/outbound interfaces for accessing and storing data. In addition, they often target to different and specific objectives. As a result, it is difficult to identify common dimensions in order to compare them. For this, we do not provide a detailed comparison of these systems in this section.

4.7.2 Progress within ElasTest

Depending on data input and the processing requirements, ElasTest can take advantage of all frameworks and database management systems defined above. Various distributed data pipelines can be designed and constructed. To cover the needs of the platform's artifacts, we create two different services. The persistence layer, under the module named EDM and the big data processing layer, under the module named EBS. The first one is always up and available to any other artifact. The second is an on-demand service, which is provisioned and de-provisioned by ESM when another service requests it. An example of such module could be the ERE with cognitive data processing. This on-demand architecture provides a cost-effective resource utilization.

Alluxio, also part of EDM, provides a data-agnostic caching layer for the persistence services of EDM. This way, the storage component also provides an agnostic layer for all consumers that do not wish to operate with concrete implementations. Such components can utilize EDM, while EDM decides where the data is actually stored or retrieved from (not yet implemented).

Both modules do not contain any technology innovation, but they offer a solid assembly that provides a stable persistence service bundle, with tested configuration that allows auto-scaling in order to facilitate the elasticity of the whole project. In addition, these two modules offer a good use case for creating portable big-data stacks bundled together and deployable in a datacenter-agnostic manner.

4.8 Dashboard Management

4.8.1 Baseline and comparative analysis

In Table 26, we present a list of dashboard management tools. Among them, the most relevant ones are: SonarQube; Grafana; Kibana; Freeboard; Checkmarx CxSAST. All these tools have in common the charts generation for dashboards, but some focus on code quality inspection while others focus on hardware metrics (like CPU usage) or log data. In the following, a detailed description of them.

Name	URL	Short Description	License
SonarQube	https://www.sonarqube.org/	SonarQube (formerly Sonar) is an open source platform for continuous inspection of code quality. It contains two dashboards that give the big picture to get hints where there might be issues and to compare projects: i) a consolidated view that shows all projects; ii) a project dashboard is also available at modules and packages level.	OSS (LGPLv3)
Squale	http://www.squale.org/	The Squale project assists developers in improving the code of their projects and helps project managers to meet quality requirements for their applications. It provides dashboards to monitor the overall health of their information system.	OSS (LGPLv3)
Geckoboard KPI	https://www.geckoboard.com	It supports metrics which indicate how a business or team is performing against its goals. They are given in terms of a target or benchmark which relates back to business's objectives and goals.	Proprietary
Checkmarx CxSAST	https://www.checkmarx.com/	It is a unique source code analysis solution that provides tools for identifying, tracking, and repairing technical and logical flaws in the source code, such as security vulnerabilities, compliance issues, and business logic problems. Dashboard Menu is available.	Proprietary
Freeboard	http://freeboard.io	It is an open source real-time dashboard builder for IoT and other web mashups.	OSS (MIT)
Grafana	https://grafana.com/	It is an open source metric analytics & visualization suite. It is most commonly used for visualizing time series data for	OSS (Apache v2)

		infrastructure and application analytics but many use it in other domains including industrial sensors, home automation, weather, and process control.	
Mozaik	http://mozaik.rocks/	It is a tool based on Node.js / react / d3 / stylus to easily craft beautiful dashboards.	OSS (MIT)
Dashbuilder	http://dashbuilder.org/	It allows for visual configuration and personalization of dashboards, and provides support for different types of visualizations using several charting libraries, full featured editor for the definition of chart visualizations.	OSS (Apache v2)
Kibana	https://www.elastic.co/products/kibana	It is a platform for analytics and visualization that allows to explore, visualize, and build dashboards on top of the log data stored in Elasticsearch clusters.	OSS (Apache v2)

Table 26. Dashboard management tools

4.8.1.1 *SonarQube*

SonarQube[77] (formerly Sonar) is an open source platform (OSS LGPLv3) for continuous inspection of code quality. It contains two dashboards: i) a consolidated view that shows all projects; ii) a project dashboard is also available at modules and packages level.
The main features of SonarQube are: i) it is an open source platform for continuous inspection of code quality; ii) it allows to create users and groups to assign them permissions; iii) it cannot interact with graphs in dashboard to filter data; iv) large community resources are available; v) there are detailed documentation on SonarQube website; vi) it supports Email alert notifications.

4.8.1.2 *Grafana*

Grafana[78] is an open source metric analytics and visualization suite. The main features of Grafana are: i) it allows for general purpose dashboard and graph composer, which runs as a web application; ii) it allows the creation of organizations and enables to add users with roles within them; iii) it's possible to interact with graphs to filter information; iv) it offers only commercial support; v) large community resources are available; vii) there are detailed documentation on Grafana website; viii) alert notifications through email, Slack, Webhook, PagerDuty, VictorOps and OpsGenie are provided; ix) it allows

[77] https://www.sonarqube.org

[78] https://grafana.com/

to add annotations into graphs; x) it allows to mix different data sources in the same graph; xi) it allows to apply ad-hoc filters.

4.8.1.3 *Kibana*

Kibana[79] is a platform for analytics and visualization. It allows to perform advanced data analysis and visualize data in a variety of types of charts, tables, and maps. It is an open source data visualization tool (OSS Apache v2) that is most commonly used in conjunction with Elasticsearch and Logstash. By default, the Kibana dashboard is public. There are no built-in role-based access (RBA) controls. It's possible to interact with graphs to filter information or change data colors. It offers only clients support whereas large community resources are available. It offers detailed documentation on its web page. Kibana needs ElastAlert plugin to alert notifications.

4.8.1.4 *Freeboard*

Freeboard[80] is an open source (OSS MIT) real-time dashboard. The main features are: i) dashboards are public on free version. On payment plans can be private; ii) it cannot interact with graphs in dashboard to filter data; iii) it does not offer support; iv) the resources of the community and documentation are poor; v) it hasn't alert notifications.

4.8.1.5 *Checkmarx CxSAST*

Checkmarx CxSAST[81] is a commercial source code analysis solution. The main features of the tool are: i) dashboard menu is available; ii) it's possible to create users and assign them roles and permissions; iii) it cannot interact with graphs in dashboard to filter data; iv) there is no solid community but Checkmarx offers documentation on its web and support to licensed customers; v) it hasn't alert notifications. Table 27 contains a summary of the functionalities available or not in each one of the above tools.

	SonarQube	Grafana	Kibana	Freeboard	Checkmarx CxSAST
Multiplatform	+	+	+	+	+
Interactive graphs	-	+	+	-	-
Role-based access	+	+	-	-	+
Open source	+	+	+	+	-
Great Community	+	+	+	-	-

[79] https://www.elastic.co/products/kibana

[80] http://freeboard.io

[81] https://www.checkmarx.com/

Support	-	+	+	-	+
Detailed Documentation	+	+	+	-	+
Alert Notifications	+	+	+/-	-	-
Annotations into graphs	-	+	-	-	-

+: Available, *-: Not Available*, *+/-: Partially available*

Common *Advantage* *Disadvantage*

Table 27. Comparison of dashboard management tools

4.8.2 Progress within ElasTest

ElasTest has a dashboard with charts generated by metrics and logs gathered during test and SUT execution. In the current version, it has the ability to interact with charts in order to offer to the users the ability to filter the data they want to focus on. SotA improvements in this area will be based on the comparison of logs and metrics for different test executions.

4.9 WebRTC Testing

4.9.1 Baseline and comparative analysis

In Table 28, we present the identified WebRTC testing tools. Among them, the most relevant ones are: TestRTC, webrtc-test, Kurento Test Framework (KTF). In the following the main features of these tools are described.

Name	URL	Short Description	License
TestRTC	http://testrtc.com/	It provides a set of tests that can be easily run by a user to help diagnose WebRTC related issues. The user can then download a report containing all the gathered information or upload the log and create a temporary link with the report result.	OSS (BSD-style)

webrtc-test	https://github.com/RestComm/webrtc-test	It is a framework for functional and Load Testing of WebRTC for RestComm (cloud communication platform for video and voice).	OSS (AGPLv3)
Kurento Test Framework (KTF)	http://dl.acm.org/citation.cfm?id=3021393	Kurento provides a high-level testing infrastructure to assess WebRTC services in terms of functionality, performance, and quality-of-experience.	OSS (LGPL license)
callstats.io : Analytics for WebRTC	https://www.callstats.io/	callstats.io keeps track of WebRTC QoE/QoS stats in real-time. For QoE, it aggregates and visualizes the Mean Opinion Scores (MOS) metrics.	Proprietary

Table 28. WebRTC testing tools

4.9.1.1 *TestRTC*

TestRTC[82] is an integrated platform aimed to test, monitor and analyze WebRTC-based communications. It has OSS (BSD-style) license. The main features of TestRTC are:

1. Use of real browsers. This is one the major strategic decisions of TestRTC: support only real web browsers as agents to assess the SUT instead of building something on top of WebRTC directly. At the time of this writing, TestRTC supports Chrome and Firefox browsers. Thanks to its own global cloud infrastructure, TestRTC allows to choose these browsers from different locations (east US, west US, Europe, and Asia);
2. JavaScript API. Developers using TestRTC can write test scripts using an API built on the top of Nightwatch.js. These tests can be directly uploaded to the TestRTC web dashboard;
3. Network awareness: TestRTC allows to configure the underlying network with custom setups, including different firewall and NAT configurations, different bitrates (static and dynamic ones), and different packet loss. To simplify this task, it provides preconfigured scenarios of the typical access networks, such as 3G (bandwidth 750Kbps, latency 250ms, packet loss 1.5%), 4G (bandwidth 4Mbps, latency 0ms, packet loss 2%), DSL (bandwidth 8Mbps, latency 40ms, packet loss 0.5%), or WiFi (bandwidth 20Mbps, latency 40ms, packet loss 0.2%), among others;

[82] http://testrtc.com/

4. Signaling protocol agnostic. TestRTC can test standards-based signaling protocols (such as SIP over WebSocket, XMPP over WebSocket, or BOSH) and proprietary ones (based on either WebSocket or HTTP(S) and REST);
5. WebRTC tests at scale. TestRTC allows to run different media session at the same time. In addition, it allows to leverage the number of concurrent users per session;
6. Monitoring. TestRTC allows to choose between different metrics and Key Performance Indicators (KPIs) such as channel types, bitrate, timing, among others;
7. WebRTC-internals analyzer. Chrome allows to download the PeerConnection updates and stats data in JSON notation. This data can be uploaded with drag-and-drop to the TestRTC web client and a detailed report about the data (jitter, frame rate, packet loss, codecs, and so on) can be provided;
8. Live preview of the remote browser tests by means of VNC (Virtual Network Computing) connections;
9. Use customizable user media for WebRTC sessions, namely VGA (video 640x480, 2948Kbps; audio: 121Kbps), HD (video 1280x720, 8069Kbps; audio: 192Kbps), Full-HD (video 1920x1080, 8069Kbps; audio: 173Kbps), only-audio (98Kbps), among others.
10. Reporting. The TestRTC dashboard shows extensive set of reports by collecting and calculating KPIs related to the voice and video streams.
11. Test history. TestRTC stores the full test execution history, allowing developers to keep tracking the evolution of the SUT in time.

4.9.1.2 *webrtc-test*

webrtc-test[83] is a low-level open source tool, based on python and Restcomm Platform (cloud communication platform for video and voice). It can be used for functional and load testing of WebRTC. It has OSS (AGPLv3) license. The main features of webrtc-test are: i) support to perform tests in Chrome and Firefox over OS GNU/Linux and OSX; ii) testing communications over Restcomm Platform; iii) it allows setup load test scenarios with python; iv) it allows to generate call traffic with any tool; iv) it allows to visualize the test results.

4.9.1.3 *Kurento Test Framework*

Kurento Testing Framework (KTF) [37] is a complete testing framework aimed to simplify the assessment of WebRTC-based applications. Kurento provides a high-level testing infrastructure to assess WebRTC services in terms of functionality, performance, and quality-of-experience. It has OSS (LGPL) license. KTF is not only for Kurento Media Server applications and can be used in general for WebRTC. The main features of KTF are: i) it automates test execution using real web browsers with Selenium; ii) it provides Test Scenarios. A scenario is the set of browsers, that can be executed on different platforms and OS, which are going to use the web application under test. KTF allows to setup

[83] https://github.com/RestComm/webrtc-test

different test scenarios based on a custom and highly customizable JSON notation; iii) it allows to configure different scopes of execution: Local Browsers, Remote Browsers (using Selenium Grid), Remote Browsers from Sauce Labs, Docker Browsers; iv) Java Support; v) it provides support for functional, performance and QoE (Quality of Experience) tests; vi) it provides the capability of monitoring a given machine, typically the sever hosting the SUT. This information can be gathered locally, i.e. the SUT is running in the same host that it is executing the performance test or remotely, i.e. the SUT is running in a different host; vii) it provides WebRTC statistics gathering for the server-side (Kurento Media Server) and also for the client-side (*peerConnection*).
Table 29 compares the main features of the above tools.

	TestRTC	webrtc-test	Kurento Test Framework
Browsers	Chrome, Firefox	Chrome, Firefox	cross-browser
Supported Platforms (testing system)	cross-platform	Linux	Windows, Linux, OS X
Supported Platforms (tested system)	Unknown.	Linux, OS X	Windows, Linux, OS X
Script Languages	Javascript	Python	Java
Graphical Tool	Yes	No	No
Distributed Testing	Yes	Yes	Yes
Reports	Yes	Yes	Yes
Media Server Dependent	No	Yes	No
		Advantage	*Disadvantage*

Table 29. Comparison of WebRTC testing tools

4.9.2 Progress within ElasTest

ElasTest User Impersonation Service (EUS) will make a progress beyond SotA in WebRTC testing. Some of the open research lines are: simulate different WebRTC network topologies; simulate interoperability issues (for example, connecting an RTP camera to a WebRTC browser); QoE applied to WebRTC testing.

4.10 Cross-browser Testing

4.10.1 Baseline and comparative analysis

Name	URL	Short Description	License
BrowserStack	https://www.browserstack.com	BrowserStack is a cloud-based tool to check web application compatibility on the web, desktop, and mobile browsers. With BrowserStack, it is possible to test web applications using online JavaScript or Selenium automated test suites.	Proprietary
Sauce Labs	https://saucelabs.com/	Sauce Labs is a cloud-hosted, web and mobile application automated testing platform.	Proprietary
Browsershots	http://browsershots.org/	Browsershots tests web page layout by taking screenshots rendered in real browsers on different operating systems.	OSS
TestingBot	https://es.testingbot.com/	TestingBot offers manual and automated cross browser testing on hundreds of browsers. With support for different flavors of native and mobile browsers, TestingBot is suitable for both web and mobile app testing.	Proprietary
Browsera	http://www.browsera.com/	Browsera is another tool to compare websites across different browsers and identify errors and differences among them.	Proprietary
CrossBrowserTesting	https://crossbrowsertesting.com/	CrossBrowserTesting is also one of the popular browser testing tools to test websites on a wide range of desktop and mobile browsers as well as remote VNC.	Proprietary
Spoon	http://square.github.io/spoon/	Spoon is a powerful and robust tool for cross-browser testing and backward compatibility.	OSS
Browserling	https://www.browserling.com/	It is a live interactive cross-browser testing service that provides effortless cross-	Proprietary

		browser testing for web developers and web designers.	
MultiBrowser	https://multibrowser.com/	It is an application targeted at cross-browser web development, especially creating/editing the CSS.	Proprietary
Microsoft Super Preview	https://www.microsoft.com/en-us/download/details.aspx?id=2020	It is a standalone tool to test web pages for cross-browser compatibility.	Proprietary
Endtest	http://endtest.io/m	It is a tool for codeless automated testing in the cloud.	Proprietary

Table 30. Cross-browser testing tools

Table 30 shows the identified cross-browser testing tools. Among them, the most popular tools are: SouceLabs, BrowserStack, EndTest and CrossBrowserTesting.
Below, they are detailed according to the type of license, the features they provide and the type of Graphical applications they test.

4.10.1.1 *Sauce Labs*

Sauce Labs[84] is a Proprietary tool that provides a cloud platform to automate the tests of web and mobile applications. The main features are: i) providing wizard to configure testing environment for manual test; ii) run parallel test executions (manual and automated); iii) recording and playback of test execution; iv) integration with Selenium automated tests; v) providing integration with JIRA issue tracker; vi) providing integration with CI Systems; vii) allowing to configure the test environment; viii) allowing reporting between CI and Sauce Labs; ix) providing Rest API; x) providing a list of test executions and actions performed in a test execution.

4.10.1.2 *BrowserStack*

BrowserStack[85] is a proprietary cloud-based tool to check web application compatibility on the web, desktop, and mobile browsers. With BrowserStack, it is possible to test web applications using online JavaScript or Selenium automated test suites.
The main features are: i) Quick Launch. It enables to create a new quick launch configuration; ii) it provides wizard to configure testing environment; iii) integration with Selenium automated tests; iv) instant access to all real mobile and desktop browsers to test the web or mobile application; v) it provides issue tracker; vi) it provides real devices for mobile testing; vii) it provides real browser installed on real machines

[84] https://saucelabs.com/

[85] https://www.browserstack.com

for web testing; viii) it enables to take rapidly a screenshot of the web application, whereas it running over many browser and platforms at the same time.

4.10.1.3 *EndTest*

EndTest[86] is a proprietary cloud platform to codeless automated testing for web applications. It supports: i) easy codeless automated testing; ii) configuration of test scenarios; iii) managing of test suites directly in the cloud; iv) keeping track of the changes on the tests.

4.10.1.4 *CrossBrowserTesting*

CrossBrowserTesting[87] provides a cloud platform to testing applications written for desktop and mobile browsers. It is also one of the popular browser testing tools to test websites on a wide range of desktop and mobile browsers as well as remote VNC.
The main features of CrossBrowserTesting are: i) it allows easy test scenario configuration; ii) it shows short information of executed tests and allows to view run test details; iii) it allows to configure automatic screenshots for different browsers and operating systems; iv) it allows to view screenshots results and compare layout differences.
Table 31 compares the above tools.

	BrowserStack	Sauce Labs	CrossBrowser Testing	EndTest
Nº. of Browsers	700+	700+	750+	90+
Languages	Python, Ruby, Java, C#, Perl, PHP, NodeJS	Python, Ruby, Java, C#, Perl, PHP, NodeJS	Python, Java, Ruby, C#	Java, Python, Ruby
Integrations	Jenkins, Travis, Circle	Jenkins, Travis, Selenium Bamboo, Circle, TeamCity	Selenium, TestComplete	Jenkins, Selenium
Mobile Browsers	Yes	Yes	Yes	Yes
Screenshot Testing	Yes	No	Yes	No
Mobile app	Yes	Yes	No	No

Table 31. Comparison of cross-browser testing tools

[86] http://endtest.io/m

[87] https://crossbrowsertesting.com/

4.10.2 Progress within ElasTest

ElasTest will advance cross browser testing tools, from a scientific and from an innovation perspective, by providing the following new features: i) simultaneous display of multiple test executions; ii) simultaneous display of the execution log from browser, JavaScript code, Web server code, etc.; iii) synchronization between video recording and logs to ease detection of failures or anomalies; iv) automatic comparison of video recordings to highlight differences between browsers/scenarios; v) integration with different SuT configurations (load, network conditions, failures, etc.).

4.11 Test Execution & Visualization

4.11.1 Baseline and comparative analysis

We present in Table 32 the identified test execution and visualization tools. Among them, the most relevant tools are: Jenkins Test Result Analyzer Plugin; Jenkins JUnit Plugin; Jenkins Test Stability Plugin, Jenkins Dashboard View Plugin; Jenkins Test in Progress Plugin; TestCollab; Rational Quality Manager; TestRail; qTestExplorer; Sauce Labs and EndTest. Below we present the main features of these tools.

Name	URL	Short Description	License
QF-Test Plugin	https://wiki.jenkins-ci.org/display/JENKINS/QF-Test+Plugin	It is a plugin that enables the integration of QF-Test with Jenkins and builds HTML reports with the results.	MIT License
Test Result Analyzer Plugin	https://wiki.jenkins-ci.org/display/JENKINS/Test+Results+Analyzer+Plugin	It is a plugin that shows in a tabular format, the history of test execution results for a job. Further, it also displays the information as a graph.	Apache License, Version 2.0
JUnit Plugin	https://wiki.jenkins-ci.org/display/JENKINS/JUnit+Plugin	The JUnit plugin provides a publisher that consumes XML test reports generated during the builds and provides some graphical visualization of the historical test results.	MIT License
Test Stability Plugin	https://wiki.jenkins-ci.org/display/JENKINS/Test+stability+plugin	This plugin adds historical information about the stability of tests.	MIT Licence

Multi-module-tests-publisher Plugin	https://plugins.jenkins.io/multi-module-tests-publisher	It is a plugin that allows publish the test execution results grouped by suite.	MIT Licence
Dashboard View Plugin	https://wiki.jenkins-ci.org/display/JENKINS/Dashboard+View	It is a plugin that provides a dashboard to display the jobs execution information.	MIT Licence
PractiTest	https://www.practitest.com/	It is an entirely SaaS end-to-end QA and Agile friendly test management tool. PractiTest helps users to manage their development and testing process, with an end-to-end approach.	Proprietary
TestCollab	https://testcollab.com/	It is a modern test management tool which offers a complete platform for application's testing.	Proprietary
IBM Rational Quality Manager	http://www-03.ibm.com/software/products/en/ratiqualmana	It is a collaborative hub for business-driven software and systems test planning.	Proprietary
TestRail	http://www.gurock.com/testrail/	It is a centralized test cases management tool.	Proprietary
OverOps	https://www.overops.com/java-monitoring	It is a static and dynamic code analysis technology to analyze code events in real time. It allows to connect the contents of the log file with the source code.	Proprietary
Wallabayjs	https://wallabyjs.com/	It is an integrated Continuous Testing Tool for JavaScript.	Proprietary
qTest Explorer	https://www.qasymphony.com/software-testing-tools/qtest-explorer/test-execution-recorder/tab/record/	It is a module of qTest Cloud Platform, to test faster, report bugs and automate test execution.	Proprietary

Sauce Labs	https://saucelabs.com/	Sauce Labs is a cloud-hosted, web and mobile application automated testing platform.	Proprietary
EndTest	http://endtest.io	It is a tool for codeless automated testing in the cloud.	Proprietary
Jenkins Test in Progress Plugin	https://wiki.jenkins.io/display/JENKINS/Test+In+Progress+Plugin	This plugin allows seeing how the tests progress during a build.	EPL 1.0

Table 32. Test execution and visualization tools

4.11.1.1 *Test Result Analyzer Plugin*

Test Result Analyzer plugin[88] provides a more visual and global view of the test results executed during successive constructions of a job. The plugin displays test execution results in three different ways:

- Displays a historical table of test results executed during successive builds of a job.
- Displays the data as a Line Chart allowing to view the test execution results trend.
- Displays the data as a Pie Chart allowing to view the test execution results distribution.

4.11.1.2 *Jenkins JUnit Plugin*

Jenkins JUnit plugin[89] allows to see the detail of each executed test and if it failed, it is possible to access to the error trace. The JUnit plugin provides a publisher that consumes XML test reports generated during the builds and provides some graphical visualization of the historical test results as well as a web UI for viewing test reports, tracking failures, and so on.
Specifically, the plugin displays the test results in two formats: i) as a trend chart; ii) as a list from which it is possible to access the details of each test.

4.11.1.3 *Jenkins Test Stability Plugin*

Jenkins Test Stability plugin[90] adds stability information to the results view of the JUnit Plugin. Specifically, it adds historical information about the stability of tests - i.e. the percentage how often they failed.

4.11.1.4 *Jenkins Dashboard View Plugin*

Jenkins Dashboard View plugin[91] displays data summary about the test results, in the same way like the previous ones, as a chart or as a table. It provides a dashboard to

[88] https://wiki.jenkins-ci.org/display/JENKINS/Test+Results+Analyzer+Plugin
[89] https://wiki.jenkins-ci.org/display/JENKINS/JUnit+Plugin
[90] https://wiki.jenkins-ci.org/display/JENKINS/Test+stability+plugin
[91] https://wiki.jenkins-ci.org/display/JENKINS/Dashboard+View

display the jobs execution information. Part of this information, it's relative to test execution results.

4.11.1.5 *Jenkins Test in Progress Plugin*

This plugin[92] provides the capability to visualize the execution logs associated to the test being executed. To be able to do this, tests needs to be modified using a custom library.

4.11.1.6 *TestCollab*

TestCollab[93] is a modern proprietary test management tool which offers a complete platform for application's testing, integration with all popular bug trackers and test automation tools, time tracking, agile methodology, requirements management, test plans and scheduling. Like some plugins of Jenkins, this management application shows a summary of statistical data about the test results.

4.11.1.7 *Rational Quality Manager*

Rational Quality Manager [94] is a proprietary collaborative hub for business-driven software and systems quality testing across virtually any platform and type of testing. This software helps teams share information seamlessly, use automation to accelerate project schedules and report on metrics for informed release decisions.
Like some Jenkins plugins, this management application shows a summary of statistical data about the test results.

4.11.1.8 *TestRail*

TestRail[95] is a centralized test case management tool to create test cases and test suites. It supports track execution and report metrics. It integrates with many issue tracking tools that make requirements from external systems to be linked to test cases in TestRail; bugs can also be created in the external systems and links can be established to the corresponding test case.
This tool shows the test historical data in a similar way than other previous described tools.

4.11.1.9 *qTest Explorer*

qTest Explorer has been already presented in the contents of GUI Automation and Impersonation tools. Please, refer for its description to Section 4.5.1.4

4.11.1.10 *Sauce Labs*

Sauce Labs has been already presented in the contents of Cross-browser testing tools. Please, refer for its description to Section 4.10.1.1.

---

[92] https://wiki.jenkins.io/display/JENKINS/Test+In+Progress+Plugin

[93] https://testcollab.com/

[94] http://www-03.ibm.com/software/products/en/ratiqualmana

[95] http://www.gurock.com/testrail/

4.11.1.11 *EndTest*

EndTest has been already presented in the contents Cross-browser Testing tools. Please, refer for its description to Section 4.10.1.3.

Table 33 shows a comparison of the tools described above:

	Statistical reports	Technical reports
Jenkins Test Result Analyzer Plugin	Test status, trend and distribution charts	No
Jenkins JUnit Plugin	Test status, trend chart	Error trace
Jenkins Test Stability Plugin	Rate stability	No
Jenkins Dashboard View Plugin	Test status, trend and distribution charts	No
Jenkins Test in Progress Plugin	No	Execution logs
TestCollab	Execution summary, distribution chart	No
Rational Quality Manager	Several charts based on different indicators (Trend, distribution, OKs vs KOs…)	No
TestRail	Distribution chart	No
qTestExplorer	No	Screenshots, Recordings and documentation
Sauce Labs	No	Screenshots, Recordings and logs
EndTest	No	Screenshots, Recordings and logs fragment

Table 33. Comparison of test execution and visualization tools

4.11.2 Progress within ElasTest

ElasTest will improve the above test execution and visualization tools both from a scientific and from an innovation perspective providing these new facilities: distributed test execution, test scenarios definitions, complete reporting (video recording, screenshots, logs, documentation). To do that, we plan to introduce some novel features, mainly related to the visualization of the execution of Tests in the large (TiLs). These features include: i) visualization of the execution of tests in the large (TiLs); ii) simultaneous playback of video recordings (on GUI tests) and logs; iii) synchronized playback of related tests to ease the comparison between them, mainly on several browsers; iv) the possibility to automatically navigate from log entry to source code to ease bug identification and fixing.

4.12 Mobile Testing

4.12.1 Baseline and comparative analysis

The most relevant tools about mobile testing are: Calabash, Monkeytalk, Robotium, Selendroid, UiAutomator, Frank, iOS Driver, eggPlant Functional, Ranorex Integrated Test Automation Tools, M-eux Test, UISpec, TOSCA Testsuite, Katalon Studio and Appium. In the following, we provide a description of their main features while Table 34 shows a comparison of these tools.

4.12.1.1 *Calabash*

Calabash[96] enables to write and execute automated acceptance tests of mobile apps. It's cross-platform, supporting Android and iOS native apps. It's also open source and free. Calabash consists of libraries that enable test code to interact programmatically with native and hybrid apps. The interaction consists of a number of end-user actions. Each action can be one of these: gestures, assertions, screenshots. Its main features are: i) Calabash supports Cucumber. Cucumber lets express the behavior of an app using natural language that can be understood by business experts and non-technical QA staff; ii) Although it's focused on Cucumber, Calabash can be used to write automated functional and acceptance tests using any Ruby-based test framework.

4.12.1.2 *Monkeytalk*

MonkeyTalk[97] is an open source mobile app automation testing tool for Android and iOS. MonkeyTalk is a simple-to-use tool which automates real, functional and interactive tests for iOS, Android, Web/HTML5, Hybrid and Flex apps. This open source tool can be used for simple 'smoke tests' or for 'data-driven test' suites on native, mobile, and hybrid apps, real devices or simulators. Its main features are: i) Open source Tool; ii) Provide Record and Playback; iii) Support both Android and iOS; iv) Support Cross platform recording; v) Easy Readable Test script; vi) Support Gestures.

[96] https://calaba.sh/

[97] https://www.perfomatix.com/blog/mobile-app-automation-testing/

4.12.1.3 *Robotium*

Robotium[98] is a test framework created to make it easy to write powerful and robust automatic black-box test cases for Android applications. With the support of Robotium, test case developers can write function, system and acceptance test scenarios, spanning multiple Android activities. Robotium has full support for Activities, Dialogs, Toasts, Menus and Context Menus. Its main features are: i) integration with Maven, Gradle or Ant to run tests as part of continuous integration; ii) Run-time binding to UI components; iii) the framework handles multiple Android activities automatically; iv) it provides an android testing framework for automating hybrid and native applications; v) it supports only Android applications; vi) it requires little knowledge on the underlaying AUT implementation.

4.12.1.4 *Selendroid*

Selendroid[99] is a test automation framework which drives off the UI of Android native and hybrid applications (apps) and the mobile web. Tests are written using the Selenium 2 client API. Selendroid can be used on emulators and real devices and can be integrated as a node into the Selenium Grid for scaling and parallel testing. Its main features are: i) full compatibility with the JSON Wire Protocol/Selenium 3 Ready; ii) no modification of app under test is required; iii) it allows testing of the mobile web using built in Android driver webview app; iv) it allows for automating native or hybrid apps; v) UI elements can be found by different locator types; vi) gestures are supported by advanced user interactions API; vii) it can interact with multiple Android devices (emulators or hardware devices) at the same time; viii) existing emulators are started automatically; ix) Selendroid supports hot plugging of hardware devices; x) it allows for full integration as a node into Selenium Grid for scaling and parallel testing; xi) it allows for multiple Android target API support (10 to 19); xii) it provides built in inspector to simplify test case development; xiii) it can be extended at runtime.

4.12.1.5 *UiAutomator*

UI Automator[100] is a UI testing framework suitable for cross-app functional UI testing across system and installed apps. The UI Automator testing framework provides a set of APIs to build UI tests that perform interactions on user apps and system apps. The UI Automator APIs allows to perform operations such as opening the settings menu or the app launcher in a test device. The UI Automator testing framework is well-suited for writing black box-style automated tests, where the test code does not rely on internal implementation details of the target app. Its main features are: i) a viewer to inspect layout hierarchy; ii) an API to retrieve state information and perform operations on the target device; iii) APIs that support cross-app UI testing.

[98] https://github.com/RobotiumTech/robotium

[99] http://selendroid.io/

[100] https://developer.android.com/training/testing/ui-automator

4.12.1.6 *Frank*

Frank[101] allows to write structured acceptance tests (using Cucumber) and execute them against iOS application. Its main features are: i) easy Setup - Getting iOS app setup for Frank should take less than 10 minutes; ii) record video - record video of test runs to show the app in action; iii) run everywhere - run the tests on both the Simulator and Device; iv) integration with CI tools.

4.12.1.7 *iOS Driver*

iOS Driver[102] automates any IOS native, hybrid, or mobile web application using the Selenium / WebDriver API. iOS-driver is fully compatible with the Selenium / WebDriver API. IOS automation is therefore as easy as automation for a browser. ios-driver fully integrates with Selenium Grid so it is possible to reuse existing web automation infrastructure including helper and utility classes (i.e. data creation, page objects etc.). Its main features are: i) it implements the JSON wire protocol; ii) it runs on emulators and devices; iii) it can be run as a regular node in a Selenium Grid; iv) it can be used in an existing Selenium Grid to manage device / simulator farm; v) it provides support for localized native apps (same test works for all the localized versions); vi) off the shelf support for iOS devices; vii) no need to jailbreak the device; viii) no additional apps needed on the device.

4.12.1.8 *eggPlant Functional*

eggPlant Functional[103] is a functional testing tool that helps test faster and easier by automating the execution of functional tests. eggPlant Functional allows test automation using its patented image-based approach to GUI testing. eggPlant is ideal for black box testing of any application and can interact with any device (including mobile, tablet, desktop, server, and the Internet of Things) in the same way a user does, by looking at the screen.

eggPlant uses sophisticated image and text search algorithms to locate objects on the screen in a completely technology agnostic manner and then drive the device. It uses advanced image analysis technology to drive and validate the system under test (SUT). Its main features are:

- Test any device, any operating system, any technology from mobile to mainframe;
- Robust record-and-playback via the eggDrive interface;
- It can be installed on Windows, Mac OSX, or Linux;
- It supports different languages including Chinese, Russian, and Japanese with full optical character recognition (OCR);
- It allows test cases involving multiple devices, using a single test script;
- It allows integration with ALM and CI tools, such as Jenkins, IBM Rational Quality Manager, and HP Quality Center for end-to-end QA test automation;

[101] http://testingwithfrank.github.io/

[102] https://ios-driver.github.io/ios-driver/

[103] https://eggplant.io/eggplant-functional/

- It allows data-driven testing availability and includes test creation GUI and debugging environment;
- It supports intuitive image-based approach and is designed for non-developer testers;
- It is technology independent, it can test any device, any operating system, and any technology;
- It provides cross-platform support, scripts reflect the UI and business logic of the application, not the code, so a single eggPlant script can deliver test automation across all platforms; e.g. Safari to Chrome, iPhone to Android;
- It allows non-intrusive testing; eggPlant does not require any modification of the application under test or jail-breaking of the underlying OS;
- eggPlant has open interfaces and an open philosophy. It can be easily integrated into any test environment and works with any continuous integration, or other test tool;
- It uses advanced image analysis technology to drive and validate the system under test (SUT).
- Finally, it is very intuitive for testers, it is entirely technology agnostic and can test any technology on any platform; from C++, to Flash, to HTML5, on mainframes and mobile devices.

4.12.1.9 *Ranorex Integrated Test Automation Tools*

Ranorex[104] is a Windows GUI test automation framework which provides seamless testing of a wide range of desktop, web and mobile applications. As every robust test automation project, it requires reliable user interface object recognition and focuses on providing the best possible recognition for all types of desktop, web and mobile software applications. Ranorex's award winning object recognition relies on smart RanoreXPath technology. RanoreXPath reduces effort in maintaining tests and at the same time simplifies the testing of dynamically built user interfaces. Its main features are: i) Record and Edit Reliable Test Actions; ii) Ranorex Repository that is used for Mapping and Test Maintenance; iii) Automated Testing of Desktop Apps, Web Apps and Mobile Apps.

4.12.1.10 *M-eux Test*

M-eux (Mobile End User Experience) Test[105] is a mobile automation product that allows companies to automate their test cases for native, web and hybrid applications on all major mobile platforms. Besides the ability to perform test automation in functionality testing, M-eux Test is also capable in performance testing and monitoring of mobile applications. Through integration with leading test management platforms, including HP ALM, Microsoft Visual Studio and IBM Rational, enterprises can re-use existing investments in testing tools and processes. It is the leading product for test automation of mobile apps running on Windows Phone, iOS, Android, BlackBerry and other mobile OS. Its main features are: i) GUI Object recognition; ii) Platform independent: M-eux Test

[104] https://www.ranorex.com/test-automation-tools/

[105] http://www.qatestingtools.com/testing-tool/m-eux-test

supports different Mobile Operation Systems; iii) M-eux Test supports Web-based applications, GUI-based applications, Wireless services and Mobile OS verifications.

4.12.1.11 *UISpec*

UISpec[106] is a Behavior Driven Development framework for the iPhone that provides a full automated testing solution that drives the actual iPhone UI. It is modeled after the very popular RSpec for Ruby. To make finding specific views in the iPhone UI easy, UISpec includes a very powerful view traversal DSL called UIQuery that allows to not only easily traverse the view heirarchy, but also to interact with the found views. Its main features are: i) a Behavior Driven Development framework for the iPhone; ii) UIQuery object contains a list of views with logic for traversing and filtering; iii) UIExpectation allows to set expectations on the UIViews; iv) it provides a simple lightweight scripting language similar to Smalltalk that supports all the functionality of UISpec; this scripting language only supports straight messages, which means there is no support for things like variables and loops or decisions (for, while, if else, etc...). And right now, UISpec recognizes only strings, integers, and Boolean values.

4.12.1.12 *TOSCA Testsuite*

TOSCA Testsuite(tm)[107] is an enterprise agile testing suite providing functional software testing tools, comprehensive test management and automation solution. TOSCA is based on LinearQ(sm) methodology which is right by design. Tosca Testsuite assists enterprises in optimizing test cases by minimizing the number of test cases needed to achieve the highest possible risk contribution for each test case. It supports combinatorial methodology and linear expansion to reduce the number of test cases and optimizes the risk coverage at the same time. Tosca Testsuite enables to assess the aggregated risk coverage from business, technical, performance and compliance perspectives. It allows: i) test management; ii) requirements and risk management in software testing; iii) readable test cases in a customizable interface; iv) quality assurance in software testing through Quality Gates; v) manual and automated software testing; vi) automated GUI and non-GUI software testing; vii) test analysis visualized by meaningful reports.

4.12.1.13 *Katalon Studio*

Katalon Studio[108] uses open-source test automation frameworks such as Selenium and Appium by eliminating their technical complexities to allow testers to efficiently setup, create, run, report and manage their automated tests. It also offers a viable alternative to commercial test automation solutions that are unaffordable to many small and medium-sized teams. Its main features are: i) it provides project templates for organizing test cases, object repository, and keywords; ii) it fully supports Web, Android, iOS and API testing on all operating systems; iii) it is easy to integrate with Jenkins, GIT, and JIRA with native plugins; iv) it generates tests automatically; v) it allows records

[106] http://www.qatestingtools.com/testing-tool/uispec

[107] http://www.online-testautomation.com/tosca-testsuite/

[108] https://www.katalon.com/

actions and generates scripts automatically using built-in keywords; vi) it allows building advanced test scripts or customizable keywords easily; vii) it allows record & playback for Mobile as well as data-driven testing.

4.12.1.14 *Appium*

Appium has been already presented in the contents of tools for GUI automation and Impersonation. Please, refer for its description to Section 4.5.1.2.

Name	Open Source	iOS	Android	Native	Hybrid	Web Apps
Calabash	+	+	+	+	+	-
Monkeytalk	+	+	+	+	+	+
Robotium	+	-	+	+	+	+
Selendroid	+	-	+	+	+	+
UiAutomator	+	-	+	+	+/-	-
Frank	+	+	-	+	-	-
iOS Driver	+	+	-	+	+	+
eggPlant Functional	-	+	+	+	+	+
Ranorex Integrated Test Automation Tools	-	+	+	+	+	+
M-eux Test	-	+	+	+	+	-
UISpec	+	+	-	-	-	-
TOSCA Testsuite	-	+	+	+	+	+
Katalon Studio	-	+	+	+	+	+
Appium	+	+	+	+	+	+

+: Available, -: Not Available, +/-: Partially available

Table 34. Comparison of mobile testing tools

4.12.2 Progress within ElasTest

After a first review of the state of the art in mobile testing tools, we decided that ElasTest will not advance in mobile testing domain right now. We plan some advancement for the remaining part of the project.

4.13 Test Management

4.13.1 Baseline and comparative analysis

The test management tools are tools used to control the lifecycle of tests in a project. Usually, these tools don't execute the test themselves, but have plugins to connect to

CI systems. Also, they usually are integrated with issue management system to report test failures. All these tools share many features between them like create test plans and test cases, run tests, generate reports, integrate with other platforms and a very advanced user management. Generally, a few steps are taken to use these tools: i) add members and assign roles; ii) create a new project; iii) create test suite and test cases; iv) run tests and view results. However, some tools lack the features that are available in the others. Table 35 shows some available test management tools, among them we describe below the most relevant ones and their features.

Name	URL	Brief Description	License
TestLink	http://testlink.org/	It is a web based test management and test execution system. It enables quality assurance teams to create and manage their test cases as well as to organize them into test plans. These test plans allow team members to execute test cases and track test results dynamically.	Open Source (GPL)
qTest	https://www.qasymphony.com/	It provides software testing and development teams with an easy to learn, easy to use, fast and scalable test management that seamlessly integrates with JIRA, and automation tools.	Proprietary
PractiTest	https://www.practitest.com/	An entirely SaaS end-to-end test management tool. PractiTest helps users to manage their development and testing process, with an end-to-end approach.	Proprietary
Zephyr	https://www.getzephyr.com/	It is a leading provider of on-demand, real-time test management solutions, offering innovative applications, seamless integrations and unparalleled, real-time visibility into the quality and status of software projects.	Proprietary
TestCaseLab	https://testcaselab.com/	It is a tool for manual QA engineers that allows to follow testing activities, creating test cases,	Proprietary

		categorizing, gathering them in test plan and starting test runs.	
EasyQA	https://geteasyqa.com/	It is a testing platform that provides full test management, test visualization, fast bug tracking system, automatic code building, build sharing, integration with GitHub, GitLab, JIRA, Youtrack, Pivotal, Redmine.	Proprietary
HP ALM Quality Center	https://saas.hpe.com/en-us/software/quality-center	It is a quality management software solution that offers software quality assurance, including requirements management, test management and business process testing for IT and application environments.	Proprietary
Squash TM	http://www.squashtest.org/en	It is the test repository manager found in the open source Squash toolkit. It enables the management of requirements as well as test cases execution in a multiproject context.	Open Source

Table 35. Test management tools

4.13.1.1 *TestLink*

TestLink[109] is a web-based test management and test execution system developed and maintained by Teamtest. It enables quality assurance teams to create and to manage test cases as well as to organize them into test plans. These test plans allow team members to execute test cases and track test results dynamically. Also, the platform offers support for processing several kinds of reports and statistics. The basic units used by TestLink are: Test Plan, Test Case, User, Test Projects and Test Specifications as described below.

***Test Plan**:* It is a basic unit for executing a set of tests on an application. Test Plans include Builds, Milestones, User assignment and Test Results. A Test Plan contains name, description, collection of chosen Test Cases, Builds, Test Results, milestones, tester assignment and priority definition. Each Test Plan is related to the current Test Project. Test Plan definition consists from title, description (html format) and status "Active" check-box.

[109] http://testlink.org/

Test Case: A test case describes a simple task in the workflow of an application. A test case is a fundamental part of TestLink. After a tester runs a test case it can either pass, fail or block it. Test cases are organized in test suites. Test Cases have the following parts: Identifier, Title, Summary, Steps, Expected results, Attachments, Importance, Execution type, and Custom fields.
User: Each TestLink user has an assigned role that defines the features available. The default types are: Guest, Test Designer, Senior tester, Tester, Leader and Administrator but custom roles can also be created.

Test Projects*:* They are the basic organizational unit of TestLink. Test Projects could be products or solutions of ther company that may change their features and functionality over time but for the most part remains the same. Test Project includes requirements documentation, Test Specification, Test Plans and specific user rights. Test Projects are independent and do not share data.

Test Specifications**:** TestLink breaks down the Test Specification structure into Test Suites and Test Cases. These levels are persisted throughout the application. One Test Project has just one Test Specification.

4.13.1.2 *qTest*

qTest[110] is a proprietary tool that provides software testing and development teams with an easy to learn, easy to use, fast, scalable test management that seamlessly integrates with JIRA, and other automation tools.
It is the most used test management tool by Agile testing teams, because it has a native integration with agile tools that allow real-time data flows to connect with testing development.

4.13.1.3 *PractiTest*

This proprietary tool[111] stands out for being a cloud-based service and, above all, for its good integration with Jira, Pivotal, Bug Trackers and other systems. PractiTest is an entirely SaaS end-to-end QA and Agile friendly Test management tool. It helps users to manage their development and testing process, with an end-to-end approach, with great JIRA and many other integrations.

4.13.1.4 *HP-ALM (quality center)*

HP-ALM quality center[112] is a proprietary quality management software solution that offers software quality assurance, including requirements management, test management and business process testing for IT and application environments. Its main features are: i) capture, edit, and track requirements; ii) create test repositories with test cases, test plans, test executions; iii) it allows the creation of test cases templates and calling test cases from other test cases; iv) it allows creation of variables inside test

[110] https://www.qasymphony.com/

[111] https://www.practitest.com/

[112] https://saas.hpe.com/en-us/software/quality-center

cases; v) it supports integrated bug tracking systems; vi) it is a good reporting tool; vii) it supports integration with other HP tools like HP-UFT, HP-QTP, HP-LoadRunner.

4.13.1.5 *EasyQA*

It is a proprietary testing platform [113] that provides full test management, test visualization, fast bug tracking system, automatic code building, build sharing, integration with GitHub, GitLab, JIRA, Youtrack, Pivotal, Redmine. It is intuitive and the user experience is very good. Also, it is the only tool with mobile support that allows to register crash reporting and automatic build for Android/iOS apps. The disadvantage is that it does not have requirements management like the other tools.

Table 36 contains a summary of the functionalities available or not in each one of the tools:

	TestLink	qTest	PractiTest	HP ALM Quality Center	EasyQA
Create project	+	+	+	+	+
Member Roles	+	+	+	+	+
Test Plan / Case	+	+	+	+	+
Requirements	+	+	+	+	-
Test Run	+	+	+	+	+
Reports	+	+	+	+	+
Bug Tracker	-	+	+	+	+
Import / Export	-	+	+	+	+
Integration	+	+	+	+	+
User experience	-	+	+	+	+
Pretty interface	-	+	+	-	+
SDK / crash reporting	-	-	-	-	+
Automatic build	-	-	-	-	+

[113] https://geteasyqa.com/

Open Source	+	-	-	-	-
			+: Available	*, -: Not Available*	*, +/-: Partially available*
			Common	*Advantage*	*Disadvantage*

Table 36. Comparison of test management tools

4.13.2 Progress within ElasTest

ElasTest is integrated with the main FOSS tool in this area, namely TestLink. We advance the state of the art by providing better error reporting when running manual tests as logs and metrics are collected automatically during test case execution. The validation of ElasTest project in the area of web testing uses this feature.

4.14 Testing Framework

4.14.1 Baseline and comparative analysis

All testing frameworks have common features like create test suites, run tests with timeout, specify parameters in tests, etc. However, each framework has specific characteristics that make it different from the others. Table 37 shows a list of testing frameworks, among them we describe below the most relevant ones classified in Java based framework (JUnit, TestNG, Spock) and JavaScript-based ones (Jasmine, Mocha, QUnit) as in the following.

Name	URL	Brief Description	License
	For Java		
JUnit	http://junit.org/junit4/	Simple unit testing framework to write repeatable tests in Java. JUnit has been important in the development of test-driven development and is one of the standard testing frameworks for Java developers.	OSS (EPLv1)
Arquillian	http://arquillian.org/	It is an integration and functional testing platform that can be used for Java middleware testing.	OSS (Apache v2)

TestNG	http://testng.org/doc/	It is a testing framework for Java inspired by JUnit and NUnit. The design goal of TestNG is to cover a wider range of test categories: unit, functional, end-to-end, integration, etc., with more powerful and easy-to-use functionalities.	OSS (Apache v2)
Jtest	https://www.parasoft.com/product/jtest/	It is an automated Java software testing and static analysis product. It includes technology for data-flow analysis, unit test cases generation and execution, static analysis, regression testing, runtime error detection, code review, and design by contract.	Proprietary
Mockito	http://mockito.org/	It is an open source testing framework for Java. It allows the creation of test double objects (mock objects) in automated unit tests for the purpose of Test-driven Development (TDD) or Behavior Driven Development (BDD).	OSS (MIT)
Spock	http://spockframework.org/	It is a testing and specification framework for Java and Groovy applications. What makes it stand out from the crowd is its beautiful and highly expressive specification language.	OSS (Apache v2)
For JavaScript			
Jasmine	https://jasmine.github.io/	It is framework independent and supports easy integration with Ruby projects and continuous builds. It allows for both DOM-less testing and asynchronous testing.	OSS (MIT)
Mocha	https://mochajs.org/	It is a feature-rich JavaScript test framework running on Node.js and in the browser, making	OSS (MIT)

		asynchronous testing simple and fun. Mocha tests run serially, allowing for flexible and accurate reporting, while mapping uncaught exceptions to the correct test cases.	
AVA	https://github.com/avajs/ava	It is a futuristic JavaScript test runner with simple test syntax. It runs tests concurrently.	OSS (MIT)
Karma	http://karma-runner.github.io/	It is a test runner for JavaScript on Node.js. It is very well suited to testing AngularJS or any other JavaScript project.	OSS (MIT)
Jest	https://facebook.github.io/jest/	It is a complete and easy to set up JavaScript testing solution. It works for any React project.	OSS (BSD)
QUnit	https://qunitjs.com/	It is a JavaScript unit testing framework. While heavily used by the jQuery Project for testing jQuery, jQuery UI and jQuery Mobile, it is a generic framework to test any JavaScript code. It supports server-side and client-side environments.	OSS (MIT)

Table 37. Testing frameworks

4.14.1.1 *Java frameworks*

Below there is a short description of the most important Java based testing frameworks.

4.14.1.2 *JUnit*

It is a unit testing framework[114] designed for Java. It is a simple unit testing framework to write repeatable tests in Java. JUnit has been important in the development of test-driven development and is one of the standard testing frameworks for Java developers. It is the most widely used framework for a long time, so you can find information, documentation and community support easily.

[114] http://junit.org/junit4/

4.14.1.3 *TestNG*

TestNG[115] is a testing framework designed for the Java programming language and is inspired by JUnit and NUnit. The design goal of TestNG is to cover a wider range of test categories: unit, functional, end-to-end, integration, etc., with more powerful and easy-to-use functionalities. Moreover, it enables to group test cases easily and supports the declaration of explicit dependencies between test methods.

4.14.1.4 *Spock*

Spock[116] is a testing and specification framework for Java and Groovy applications. What makes it stand out from the crowd is its beautiful and highly expressive specification language. It has been written in Groovy, which is less verbose than Java. It supports mocking and stubbing.

Table 38 contains a summary of the functionalities available or not in each one of the above frameworks.

	JUnit	TestNG	Spock
Open Source	+	+	+
Timeout Test	+	+	+
Suite Test	+	+	+
Parameterized test	+	+	+
Annotation Support	+	+	+
Group Test	-	+	-
Dependency Testing	-	+	-
Mocking	-	-	+
Stubbing	-	-	+
Know an extra language	-	-	+
	+Available	*-: Not Available*	*, +/-: Partially available*
	Common	*Advantage*	*Disadvantage*

Table 38. Comparison of java testing frameworks

[115] http://testng.org/doc/

[116] http://spockframework.org/

4.14.1.5 *JavaScript frameworks*

Below there is a description of the most important JavaScript based frameworks. Table 39 contains a summary of the functionalities available or not in each one of the above tools. Of the three frameworks, it should be noted that Jasmine and Mocha are very similar since they share many aspects such as syntax, that is easier than QUnit syntax.

4.14.1.6 *Jasmine*

It is an open source[117] testing framework for JavaScript. It comes with a built-in assertion library. It allows easy integration with Ruby projects and continuous builds. It allows for both DOM-less testing and asynchronous testing. It includes a descriptive syntax for BDD paradigm and allows simple setup. It is supported by many CI servers (TeamCity, Codeship, etc.) and some that do not support natively have plugins (Jenkins has a maven plugin).

4.14.1.7 *Mocha*

Mocha[118] is an open source framework for testing code. It is a feature-rich JavaScript test framework running on Node.js and in the browser, making asynchronous testing simple and fun. Mocha tests run serially, allowing for flexible and accurate reporting, while mapping uncaught exceptions to the correct test cases. Mocha does not come with built in mocking, it's necessary to use another library called Sinon. Mocha does not come with a built-in assertion library but there are some libraries available like Chai. It has aliases for functions to be more BDD-oriented or TDD-oriented. It allows for simple setup and is supported by some CI servers and plugins.

4.14.1.8 *QUnit*

QUnit[119] is an open source framework of JavaScript unit testing that helps us to test the code. While heavily used by the jQuery Project for testing jQuery, jQuery UI and jQuery Mobile, it is a generic framework to test any JavaScript code. It supports server-side and client-side environments. Assertion library is included. Like Mocha, QUnit doesn't come with built in mocking and needs another library like Sinon. Asynchronous testing can be complicated. It allows lots of support across the board, from QA to CI server support.

	Jasmine	Mocha	QUnit
Easy Asynchronous Test	-	+	-
Easy syntax	+	+	-
Easy configuration	+	+	-
BDD friendly	+	+	-
Good Support	+	-	+
Assertions	+	-	+

[117] https://jasmine.github.io/

[118] https://mochajs.org/

[119] https://qunitjs.com

Mocking	+	-	-
Open Source	+	+	+
	+Available	*-: Not Available*	*, +/-: Partially available*
	Common	*Advantage*	*Disadvantage*

Table 39. Comparison of JavaScript testing frameworks

4.14.2 Progress within ElasTest

ElasTest is going to progress on the reusability of tests by means of the ElasTest Orchestration Engine. As part of this work, it is possible to provide to testers a reference implementation as JUnit 5 extension. This extension will allow for easy implementation of tests, reusing the test code in different tests cases and easy parameterization of tests. In addition, extensions for testing frameworks will be provided for several Test Support Services (TSS).

4.15 Virtualization

4.15.1 Baseline and comparative analysis

Virtualization as platform/hardware virtualization refers to the creation of a virtual machine/container that acts like a real computer with an operating system. Software executed on these virtual machines is separated from the underlying hardware resources. For example, a computer that is running Microsoft Windows may host a virtual machine that looks like a computer with the Ubuntu Linux operating system; Ubuntu-based software can be run on the virtual machine.
The following characteristics will be used for proper comparison of the proposed solutions:
Model of abstraction: Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS), Container-as-a-Service (CaaS).
Resource Virtualization: Virtual Machines, Containers.
License*:* OSS, Proprietary.
Management*:* API, CLI, SDK, Dashboard.

Table 40 shows a list of the most relevant virtualization tools that are described below.

Name	URL	Brief Description	License
KVM	https://linux-kvm.org	Kernel-based Virtual Machine (KVM) is a virtualization infrastructure for the Linux kernel that turns it into a hypervisor.	Freeware

WMWare	https://vmware.org	VMware Workstation Pro and VMware Workstation Player are the industry standard for running multiple operating systems as virtual machines on a single PC.	Licensed
Docker	https://www.docker.com/	Docker enables the execution of applications as a lightweight container.	OSS (Apache 2.0)
LXD	https://linuxcontainers.org/lxd/	LXD is a container-based hypervisor.	OSS (Apache 2.0)

Table 40. Virtualization tools

4.15.1.1 *KVM*

Kernel-based Virtual Machine (KVM)[120] is a virtualization infrastructure for the Linux kernel that turns it into a hypervisor. KVM requires a processor with hardware virtualization extensions.

Resource Virtualisation. By itself, KVM does not perform any emulation. Instead, it exposes the /dev/kvm interface, which a user space host can then use to: i) Set up the guest VM's address space. The host must also supply a firmware image (usually a custom BIOS when emulating PCs) that the guest can use to bootstrap into its main OS.; ii) Feed the guest simulated I/O; iii) Map the guest's video display back onto the host.

On Linux, QEMU versions 0.10.1 and later is one such user space host. QEMU uses KVM when available to virtualize guests at near-native speeds, but otherwise falls back to software-only emulation. Internally, KVM uses SeaBIOS as an open source implementation of a 16-bit x86 BIOS.

License*:* KVM's parts are licensed under various GNU licenses: KVM kernel module: GPL v2; KVM user module: LGPL v2; QEMU virtual CPU core library (libqemu.a) and QEMU PC system emulator: LGPL; Linux user mode QEMU emulator: GPL.

Management: KVM supports a series of graphical management tools: i) Kimchi – web-based virtualization management tool for KVM; ii) Virtual Machine Manager – supports creating, editing, starting, and stopping KVM-based virtual machines, as well as live or cold drag-and-drop migration of VMs between hosts; iii) Proxmox Virtual Environment – an open-source virtualization management package including KVM and OpenVZ. It has a bare-metal installer, a web-based remote management GUI, and optional commercial support; iv) OpenQRM – management platform for managing heterogeneous data center infrastructures; v) GNOME Boxes – Gnome interface for managing libvirt guests on Linux; vi) oVirt – open-source virtualization management tool for KVM built on top of libvirt.

[120] https://linux-kvm.org

4.15.1.2 *VMware*

VMware Workstation[121] is a hosted hypervisor that runs on x64 versions of Windows and Linux operating systems; it enables users to set up virtual machines (VMs) on a single physical machine and use them simultaneously along with the actual machine. Each virtual machine can execute its own operating system, including versions of Microsoft Windows, Linux, BSD, and MS-DOS.
VMware Workstation supports bridging existing host network adapters and sharing physical disk drives and USB devices with a virtual machine. It can simulate disk drives; an ISO image file can be mounted as a virtual optical disc drive, and virtual hard disk drives are implemented as vmdk files.
Resource Virtualisation: VMware Tools, a package with drivers and other software available for the various guest operating systems. It has several components, including the following: i) Drivers for emulated hardware; ii) VESA-compliant graphics for the guest machine to access high screen resolutions; iii) Network drivers for the vmxnet2 and vmxnet3 NIC Ensoniq AudioPCI audio; iv) Mouse integration; v) Support of shared folders and drag-and-drop file transfer between host and guest. This functionality is described as HGFS (Host Guest File System), and may be disabled by default for security; it may be enabled by changes to the VMX configuration file; vi) Clipboard sharing between host and guest; vii) Time-synchronization capabilities (guest synchronizes with host machine's clock); viii) Support for Unity, a feature that allows seamless integration of applications with the host desktop by hiding virtual monitor and drawing the windows of applications running in the virtual machine on the host; ix) Unity support was added for Windows 10 and removed for Linux in Workstation 12.
License*:* VMware Workstation is developed and sold by VMware, Inc., a division of Dell Technologies. There is a free-of-charge version, VMware Workstation Player, for non-commercial use. An operating systems license is needed to use proprietary ones such as Windows. Ready-made Linux VMs set up for different purposes are available from several sources.
Management: Besides the command line tool, VMWare provides a GUI. VMware Workstation Pro can save the state of a virtual machine (a "snapshot") at any instant. These snapshots can later be restored, effectively returning the virtual machine to the saved state, as it was and free from any post-snapshot damage to the VM.
VMware Workstation includes the ability to group multiple virtual machines in an inventory folder. The machines in such a folder can then be powered on and powered off as a single object, useful for testing complex client-server environments.
Docker and LXD are described in Section 4.6.1.3 ans Section 4.6.1.4 respectively.

Table 41 shows a comparison of virtualization tools.

[121] https://vmware.org

	KVM	Docker	LXD	VMware
Model of abstraction	IaaS	CaaS	CaaS	IaaS
Type of virtualized resources	VM	Container	Container	VM
License	OSS (Apache 2.0)	OSS (Apache 2.0)	OSS (Apache 2.0)	Proprietary
Management	RESTful API, GUI	CLI, GUI, SDKs	RESTful API, CLI, GUI, SDKs	RESTful API, GUI

Table 41. Comparison of virtualization tools

4.15.2 Progress within ElasTest

ElasTest platform will be compatible with the most popular virtualization solutions listed above.
For the initial stage of the project, all the ElasTest components will be initially deployed and tested on Docker Containers. As the platform aims to address SuTs that may be deployed using KVM or VMware, adaptors will be created in order for these to be supported at both orchestration and application level.

5 Mapping of ElasTest components to SotA advancements

Table 42 reports a mapping between the ElasTest components and the several aspects of the SotA that have been identified so far. Specifically, each cell of the table results marked if a specific component of the Elastest architecture (on the column) enhances the SotA with respect to the referred aspect (on the row). Notably, if the Elastest component only uses solution from the SotA but it does not contribute with new ones, thus the cell is not marked. Finally, each cell of the second column of Table 42 refers to the section where the progress within Elastest is detailed for that specific aspect.

For the sake of completeness, a detailed description of the architecture and its components we refer to Deliverable D2.3 ElasTest requirements use-cases and architecture v1 [39].

SotA Aspect	Progress Within ElasTest	ElasTest Components													
		ETM	EDM	ESM	EIM	EPM	EMP	EUS	EDS	ESS	EMS	EBS	ERE	EQE	EOE	ECE
Continuous Integration	ElasTest progress is in Section 4.1.5					Y										
Non-functional Testing	ElasTest progress is in Section 4.2.3	Y			Y	Y					Y				Y	
Security testing	ElasTest progress is in Section 4.3.2	Y								Y						
Monitoring	ElasTest progress is in Section 4.4.2			Y	Y	Y	Y				Y					Y
GUI automation and impersonation	ElasTest progress is							Y	Y							

	in Section 4.5.3									
Machine learning/Recommender system applied to testing	ElasTest progress is in Section 3.3.6							Y	Y	
WebRTC Testing	ElasTest progress is in Section 4.9.2	Y			Y	Y				Y
Cross-browser Testing	ElasTest progress is in Section 4.10.2	Y	Y		Y					
Test Execution & Visualization	ElasTest progress is in Section 4.11.2	Y								Y
Mobile Testing	ElasTest progress is in Section 4.12.2 4.1.5				Y					

Test Management	ElasTest progress is in Section 4.13.2	Y						Y								
Testing Framework	ElasTest progress is in Section 4.14.2	Y						Y	Y	Y	Y	Y			Y	
Virtualization	ElasTest progress is in Section 4.15.2			Y		Y	Y					Y				
Test Orchestration	ElasTest progress is in Section 3.2.3														Y	
Cloud Instrumentation	ElasTest progress is in Section 4.6.2				Y	Y	Y									Y
Data Ingestion	ElasTest progress is in Section 4.7.2		Y									Y				

Dashboard Management	ElasTest progress is in Section 4.8.2	Y

Table 42. ElasTest components vs SotA advancements

6 Market Analysis

This section is about the market analysis which is a quantitative and qualitative assessment of a market. It looks into the size of the market both in volume and in value, as well as the various customer segments and the key competition. This section outlines and describes all these aspects [33].

The goal is to analyze IT trends and Market trends and the context in various IT areas in which ElasTest can later create impact. There are several markets and areas in which ElasTest provides impact, which, therefore, must be analyzed and followed during the project life. In order to keep the project development aligned with the needs, business requirements and trends among the end-users and other defined stakeholders of ElasTest ecosystem, it is necessary to have a general overview of highly-related markets such as IT Market and Cloud Market, DevOps and the DevOps continuous integration methodologies.

Some tendencies we found during our research related to testing activities are: there's a tendency to automatize testing in general and specifically in regression testing, because test automation and automated functional testing has the capability to decrease the overall cost of testing and improve software quality enabling the test team to focus on deeper aspects of testing. There will be an increase in testing efforts in upcoming years, but at the same time testing will also become smarter, important problems are: how to run tests, what parts need to be executed and tested? What tools should be used? Should DevOps be implemented? etc., In fact, developer's uses mostly focus on continuous integration to ensure the best quality of the software. Also, there's an incipient use of the Agile methodologies and a clear tendency for Test Driven Development (TDD) methodology that is being mostly widely adopted. We will discuss all these tendencies later on, while in the next section we will focus on main market trends.

6.1 IT market

The IT market [34] is expected to be continuously growing. The software industry is one of the key drivers in European economy. According to the Gartner Consulting[122] firm, the software market has been immersed during the last years in a transition provoking disruptive changes in how software services and technologies are developed, deployed, accessed and used. Gartner forecasts that enterprise software continues to exhibit strong growth, with worldwide software spending projected to grow 9.5% during this year 2018, and it will grow another 8.4% percent in 2019 to total $421 thousand million. Organizations are expected to increase spending on enterprise application software in 2018, with more of the budget shifting to software as a service (SaaS). Companies will continue to invest in IT as they anticipate revenue growth, but their spending patterns

[122] http://www.gartner.com/technology/research/it-spending-forecast

will shift. In terms of categories, the Gartner Worldwide IT Spending Forecast[123], predicts the following indicator behaviours for various major technology trends across the devices, IT services, data center systems, enterprise software, communication and telecom services markets, and across geographies. For the next period, Gartner predicted that worldwide IT spending was total $3.52 million million in 2017, will be surpassed during 2019 in $3.78 million million (see Table 43).

	2017 Spending	2017 Growth (%)	2018 Spending	2018 Growth (%)	2019 Spending	2019 Growth (%)
Data Center Systems	178	4.4	179	0.6	179	-0.2
Enterprise Software	355	8.9	389	9.5	421	8.4
Devices	667	5.7	704	5.6	710	0,9
IT services	933	4.3	985	5.5	1.30	4.6
Communications	1.393	1,3	1,427	2,4	1,443	1.1
Overall IT	3,527	3.8	3,683	4.5	3,784	2,7

Table 43. Forecast summary (billions of US dollars), 2018-2019

Based on Gartner's forecast for worldwide dollar-valued IT spends, we should highlight the forecast segment software: the first quarter of 2018 considers a positive 5.6% growth. Other segments also continue to grow due to various reasons. This has an impact in the IT Services market expenditure, which is expected to return to growth in 2018, up to 5.5% in 2018 and is expected to reach 1.030 thousand million in 2019. The spending figures on IT show a tendency that cannot be ignored, therefore an increase on investments and revenues for the IT market is ensured.

Again, from Gartner analysis, we should highlight that only two segments are forecasted to increase: the worldwide IT services market and the digital business, intelligent automation, and services optimization and innovation that continue to drive growth in the IT global market[124], but should be also watched and will depend on broad economic challenges.

Other segments continue to change due to various reasons, such as the one detected as cloud shift. One of the main reasons is the accelerating momentum in cloud infrastructure adoption and buyer acceptance of the cloud model. IT spending is progressively shifting from traditional and well known IT offerings to cloud services. This impacts in the IT Services market expenditure, because the figures for cloud shift spent

[123] http://www.gartner.com/technology/research/it-spending-forecast

[124] https://www.gartner.com/smarterwithgartner/gartners-top-10-technology-trends-2017/

in 2016 were estimated to reach $111 billion, increasing to $216 billion in 2020[125]. Noting that, cloud shift rates are determined by comparing IT spending on cloud services with traditional non-cloud services in the same market categories. With respect to these data, the solution envisioned by the Elastest project is a cloud-based platform that can be offered on the IT market in order to enable/ease the creation of a new generation of software testing services.

Another report from Market and Markets [35], suggests that the Internet of Things (IoT) testing market size is estimated to grow from $ 302.9 Million in 2016 to $ 1,378.5 Million by 2021, at a Compound Annual Growth Rate (CAGR) of 35.4% from 2016 to 2021. They suggest that the major forces driving for example the IoT Testing Market include the growing need for Internet Protocol (IP) testing of the increasing number of IoT devices and applications. Also in this respect, the Elastest project aims to solve actual needs from the IT market. Specifically, some of the technologies as well as some of the verticals developed within the project address the testing of IoT solutions.

Furthermore, the report in [43] highlights the rising importance of Continuous Integration, DevOps, and Agile as factors which are driving the market and continue to grow in adoption with quality assurance (QA) as well. As better detailed in the deliverables D2.3 [39] D3.1 [40] D6.1 [41] and D6.2 [42] the technical solution promoted by the Elastest project has been natively conceived in order to be integrated with any Agile process, Continuous Integration platforms, and with DevOps practices as well.

Another study done by CapGemini [36], the industry's largest research study and analysis of testing and quality assurance trends, explains that there are three points that make it to the "Top 3 list" of both Priorities of QA & Testing and Aspects of IT strategy. These three are: i) enhance security; ii) enhance customer experience; iii) higher quality of software solutions. The study shows that the budgets for software testing and quality assurance grew 40% (forty percent) in 2017 compared to 2016. Stronger focus on QA may not be seen as a new finding; however, it is true that a growth of 40 % is a lot. We must note that the whole industry is growing fast. Nevertheless, by 2019 it is expected, that testing and QA will take a 40 % amount of total IT budgets worldwide, clearly showing that companies do understand the importance of software testing. The digital transformation continues to drive IT strategy of firms and this is linked to the QA and testing functions.

Clearly, there are challenges around managing the immense costs that implies the test environment management within firms. As such, there's a need to have an efficient strategy and in each level of the QA and testing activities.

As demonstrated in other domains (e.g. mobile O.S.), the availability of an active community such as the one promoted by the Elastest project, and that it works on an open solution for cloud platform for software testing should contribute to: the definition of reliable and reusable testing component/services; the improvement on the efficacy of the testing-based QA procedures; reduce the overall costs of software testing activities in the long term.

[125] http://www.gartner.com/newsroom/id/3384720

6.2 Cloud Market and cloud service categories

Goldman Sachs in their study suggests that spends on cloud computing infrastructure and platforms will grow at a 30% CAGR from 2013 through 2018 compared with 5% growth for the overall enterprise IT[126]. And even indicates that Amazon, as the big player, will reach a share of 26% of the IaaS and PaaS markets ($4B in revenue). Another study, from the firm IDC[127] forecasts that public cloud spending will more than double to 127.5 billion dollars by 2018, with $24.6 billion for IaaS; $20.3 billion in PaaS expenditures and $82.7 billion in SaaS expenditure. In other studies, for example, Cisco provides a comparative analysis considering the year 2015-2020 and revises three markets: IaaS, PaaS and SaaS.

In terms of types of cloud, IDC FutureScape[128] predicted that during 2017 and onwards various versions of cloud computing as a new delivery model is preferred by companies, and there will be a 11% shift of IT budget away from traditional in-house IT delivery. Also, the study estimates that the 27.8% of the worldwide enterprise applications market will be SaaS-based, generating $50.8B in revenue up from $22.6B or 16.6% of the market back in 2013. Another relevant figure is that 35% of new applications will use cloud-enabled, continuous delivery and enabled by faster DevOps life cycles to streamline time-to-market and more agile business models and innovation pace.

Considering now QA and testing figures, the IT spending allocated to ensuring application quality and performance will rise reaching 40% by 2018. All people now use Apps, and are demanding higher quality as these are becoming more sophisticated, and offering various services. Due to the demand for high quality products and the IT trends such as Artificial intelligence, BigData analytics, cloud, mobility, virtualization, etc., testing to ensure QA has become more than just a need. This pushes organizations towards allocating increased investment of their IT budget (around 40%) for software testing and QA as shown in Figure 21**.**

[126] https://www.forbes.com/sites/louiscolumbus/2015/01/24/roundup-of-cloud-computing-forecasts-and-market-estimates-2015/#1501b5eedb7a

[127] http://innovativeii.com/2015-2017-forecast-cloud-computing-skyrocket-rule-delivery/

[128] https://www.idc.com/research/Predictions15

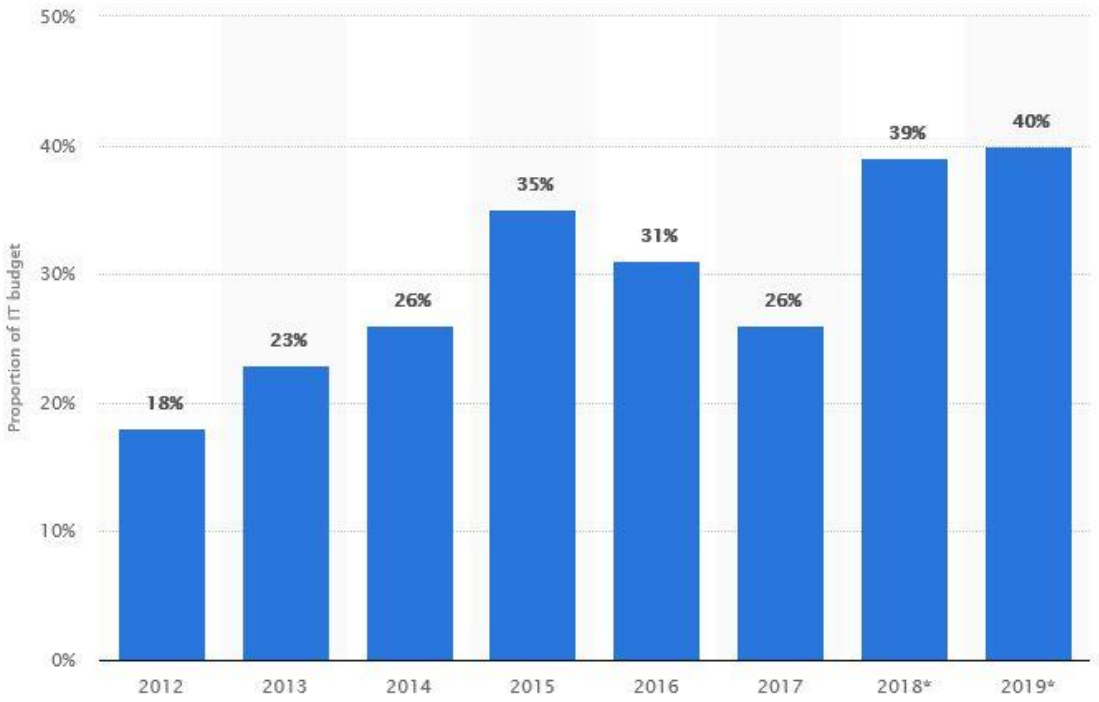

Figure 21. Percentage and port of the total IT budget allocated to QA and testing

Another study by Capgemini [129] World Quality Report 2017-18, and surveyed participants from all over the world, found that they have seen an increase in the proportion of IT budgets spent on QA and testing over the last four years, up from 63% in 2016. The study also shows that during 2017, QA and testing budget spent on personnel was 5% back in 2014, but in 2015 it grew up to 33%, and for 2020 they foresee 32%, as shown in Figure 22, where back in 2015 the investment raised up to 35%, in 2016 31%, in 2017 went down to 26%. But why are these figures decreasing?

[129] World Quality Report 2016-17, CapGemini. CapGemini own publication. www.de.capgemini.com

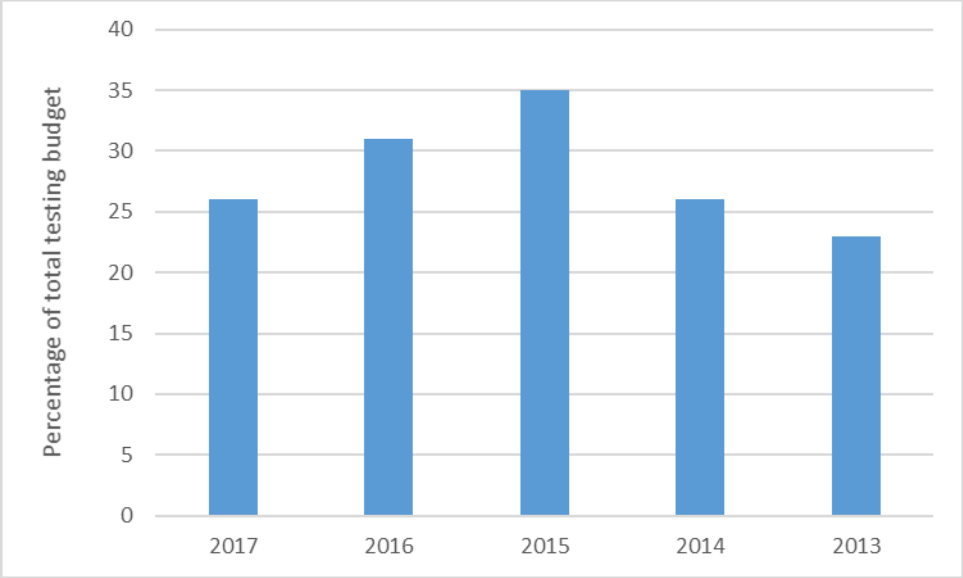

Figure 22. Percentage and port of the total IT budget allocated to testing per year

The reasons the Capgemini suggests is that manual testing is still too costly; there's a need to increase budget allocated in firms dedicated to QA and testing areas. Indeed, 39% of respondents say that manual testing is the most important and top one technological challenge in application development process within organizations. This decrease in investment may be a good sign for organizations, suggesting that maybe is because these are achieving greater efficiency over time. Also, this is because Agile and DevOps teams are now part of business units, and maybe difficult to consider and account testing activities in this respect. Therefore, let's revise and understand how Agile and DevOps methodologies fit in the testing scenario.

During 2015, we saw the acceptance of testing as an early activity in the software development lifecycle. This was predominantly due to the widespread adoption of Agile and DevOps methodologies by organizations across the globe, all aiming to produce fast applications and reach the market[130].

6.3 Devops

DevOps programming environments, supporting developers of critical system applications, including complex distributed systems, especially in the context of adaptive cloud applications, are starting to become ever important. According to a study from analyst firm Gartner[131], spending on cloud computing infrastructure and platforms will grow at a 30% CAGR from 2013 through 2018 compared with 5% growth for the overall enterprise IT, with Amazon as the big player with shares taking 26% of the IaaS and PaaS markets ($4B in revenue). According to IDC forecasts (Enterprise tech) public cloud spending will more than double to 127.5 billion dollars by 2018, with $24.6 billion for IaaS; $20.3 billion in PaaS expenditures and $82.7 billion in SaaS

[130] https://www.gartner.com/newsroom/id/2999017

[131] https://www.gartner.com/newsroom/id/3815165

expenditure[132]. Another analysis from Cisco provides a comparative analysis of IaaS, PaaS and SaaS forecasts from 2013 to 2018 that predicts that 59% of the total cloud workloads will be Software-as-a-Service (SaaS) workloads, up from 41% in 2013, eating space from IaaS and PaaS[133] (Figure 23). In terms of types of cloud, IDC FutureScape predicts that by 2017, *"there will be an 11% shift of IT budget away from traditional in-house IT delivery, toward various versions of cloud computing as a new delivery model"*, estimating that the 27.8% of the worldwide enterprise applications market will be SaaS-based, generating $50.8B in revenue up from $22.6B[134]. Another relevant figure is that 35% of new applications will use cloud-enabled, continuous delivery and enabled by faster DevOps life cycles to streamline time-to-market and more agile business models and innovation pace by 2018.

Figure 23. SaaS most highly deployed global cloud service by 2018

The DevOps movement has taken on a lot of relevance in the past years, integrating development and operations teams to drive faster and more efficient applications to the business, and to customers. Even now some suggest is not a market anymore, is more a "tool centric-philosophy".

> *"Organizations must be proactive in breaking down silos between business, development, quality and operations, and use agile and DevOps to focus on customer value and business relevance."*[135]

[132] https://www.idc.com/getdoc.jsp?containerId=prUS43511618

[133] https://www.cisco.com/c/en/us/solutions/collateral/service-provider/global-cloud-index-gci/white-paper-c11-738085.html

[134] https://www.idc.com/research/container_error.jsp

[135] https://www.capgemini.com/news/world-quality-report-2016-shows-business-operations-at-risk-with-68-per-cent-of-organizations/

This was said by Hans van Waayenburg, Member of the Group Executive Committee and Leader of the Testing Global Service Line, Capgemini Group. In this line, also a report from Gartner explains:

> *"DevOps, in itself, is not a new concept. A development-to-operations lifecycle has existed for quite some time. The latest developments include the ambition to industrialize, automate, and connect the entire process covering infrastructure, application, as well as business changes. The prime focus is on outage reduction and quality improvement"*[136].

According to a report of Market and Markets[137], DevOps market will grow from USD 2.90 Billion in 2017 to USD 10.31 Billion by 2023, at a Compound Annual Growth Rate (CAGR) of 24.7% during the forecast period. The demand for DevOps solutions and services among enterprises is expected to gain huge traction, due to the increasing need for fast application delivery with high quality. The base year considered for this study is 2017 and the forecast period is 2018–2023. Another study from Gartner [138]shows that the benefits of DevOps are that: IT organizations can drive faster delivery of features, continuous software delivery, and other advantages derived from successfully merging development, QA and IT operations. Rather than being a market per se, DevOps is a philosophy, a cultural shift that merges operations with development and demands technologies to facilitate collaborative change. Gartner predicts a growth of 21% in DevOps strategy, and currently it's used by 25 percent of Global 2000 organizations worldwide.

Gartner views DevOps as a virtual (and likely temporal) market and has focused the scope of the definition on tools that support DevOps and practices associated with it in the context of continuous delivery, continuous improvement, infrastructure and configuration as code, and so on. Gartner categorizes these tools as DevOps-ready, -enabled and -capable tools.

Knowing how to operate software and services to deliver a good customer experience is critical to organizations. This is why the DevOps approach has been widely accepted because it takes into account continuous assessment. DevOps is positioned right in the Peak of Inflated Expectations, in Gartner's latest Hype Cycle for application services, along with other key trends that are included in the present analysis.

DevOps means cooperation among: the developers, quality professionals, and IT professionals, this is basically an ideology based on continuous collaboration and integration between the different departments of an IT Organization. In this scenario, testing plays the most important business critical role as developers are involved not just in the correctness of their code, but also in the testing their software developed and

[135]https://www.de.capgemini.com/resource-file-access/resource/pdf/devops_pov_2015-12-18_final_3.pdf

[137]https://www.marketsandmarkets.com/Market-Reports/devops-824.html?gclid=CjwKCAjwlcXXBRBhEiwApfHGTbAsx9OSG7jNKqrurMQY7HkEmaDZs5Hm_S_cJIOlpkOY3-6e0A6XHBoCf8EQAvD_BwE

[138]https://www.gartner.com/newsroom/id/2999017

overall quality of the resulting application/code. DevOps thus is forcing businesses towards greater speeds of deployment and quality assurance and is thus helping them realize higher returns on investment and faster time to market in a cost-efficient manner. Also, organizations need to provide the best experience to users (QoE) in making organizations change their strategy in IT and focusing on testing. They are now moving away from just providing performance tests to providing constant testing mechanisms to ensure that software quality meets user needs.

Since few years ago, the acceptance of testing as an early activity in the software development lifecycle was widely known. This was predominantly due to the widespread adoption of DevOps and Agile methodologies by organizations across the globe and all efforts aiming to shortening their software delivery cycles and producing fast applications to reach the market.

6.4 Continuous integration

Continuous integration (CI)[139] is widely know now by all developers and companies because it is the practice of regularly building and testing all the changes done to the software. The ideal is that such changes and this process are done automatically and should be as early as possible. Continuous integration is usually accomplished through automation of testing. By automating tests, users can ensure that the most important features of their applications are working, regardless of the changes that developers make to the code. The process of CI usually allows experimenting and implementing new features and finally releasing updates very fast. Moreover, continuous integration systems are used by most developers to find errors in their code early on in a project or in a given sprint, which can save time and prevent delays in the delivery of new features. Continuous integration systems are an early part of the DevOps workflow that is solely defined by developers or companies that are providing software. Other areas that are important in this process are continuous delivery and continuous deployment (and configuration management). Usually, the continuous integration system products have the following features:

- Allow developers to execute automatically the unit tests;
- Perform automated tests against new code;
- Show a list of tests that have passed and failed;
- Perform all the necessary actions to create a fully functioning build of the software when all tests have passed.

The authors of [43] analysed the top “Continuous testing service providers”. They have identified the 11 most significant ones according to their revenue, clients and some parameters described in the study, such as providing continuous testing services and a proof that those vendors provided a majority of testing services that work in an Agile and Devops context. They found the following ones: Atos, Cigniti Technologies, EPAM Systems, Hexaware Technologies, HCL Technologies, Larsen & toubro infotech (LTI),

[139] https://www.g2crowd.com/categories/continuous-delivery

Mindtree, NIIT Technologies, QualiTest, Syntel, and Tech Mahindra as shown in Figure 24.

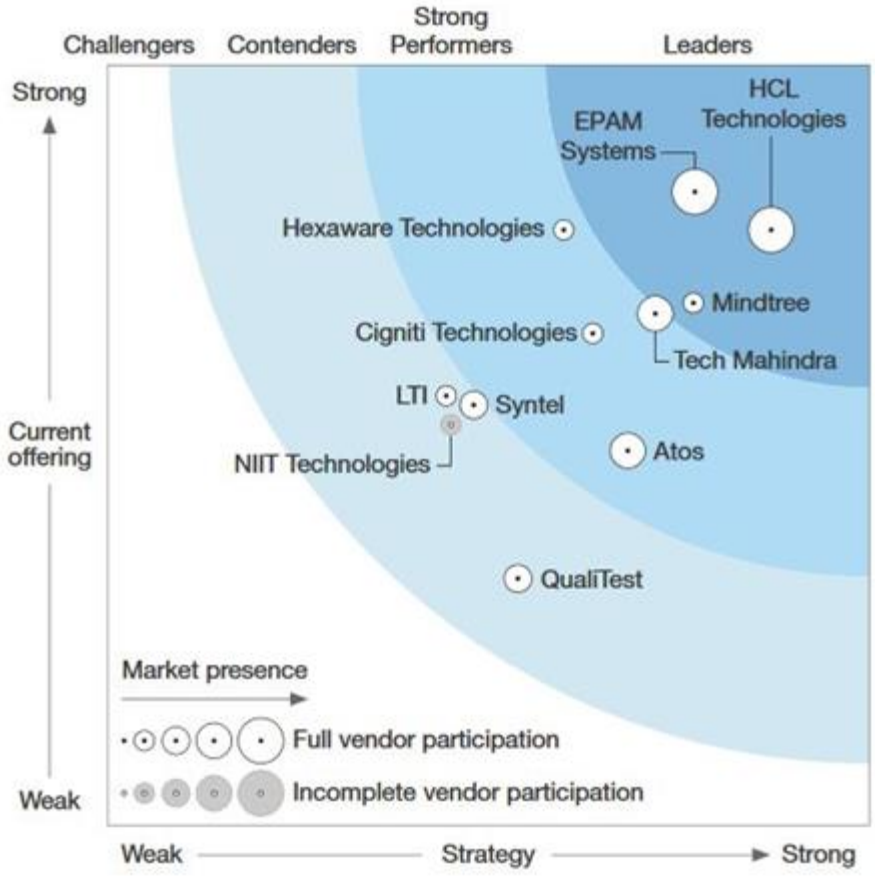

Figure 24. Top continuous testing service providers (from [54])

Among other findings, they suggest that Test Centers of Excellence in organizations are over. These organizations are not looking (as in the past) at having centralized Test Centers of Excellence anymore. This is because the test automation developers today belong to the Agile teams which are also best and suitable organizations to meet the market demands. Now, the reality is that the former testing areas in companies are now making a change towards quality engineering and are also part of the continuous integration process. Also, testing within key organizations and main service providers is intended to become more an iterative and changing process and at the same time it will be integrated with development.

6.5 Software Testing and key vendors

According to the analyst firm Nelson Hall, and by the Software Testing NEAT reports, (NelsonHall Vendor Evaluation and Assessment)[140] from November 2017, it is evident that the overall software testing market size was about $34 BN by last year 2017. The

[140] https://research.nelson-hall.com/sourcing-expertise/neat-reports/

firm Gartner predicts that the worldwide discrete software testing market spending is to be increased by 14 % CAGR with product testing growing at the rate of 9.1 % and application testing at 15.3 % when considering application testing, which acquires around 90 % of the software testing market services. On the other hand, when talking about traditional vs specialized testing services, the traditional testing services spent is getting transferred more and more to specialized testing spent (and to different specialized skilled people) and that is because the disruption in technologies and thus leading to faster time to market. For example, they predict that by 2020, intelligent automation, including the use of artificial intelligence (AI) and machine learning technologies, will be among the top three selection criteria in 30% of application testing engagements. According to the Magic Quadrant for Application Testing Services[141], a study from last November, suggests that by 2020, 60% of testing resources will need to have a combination of testing skills, application development skills, and business process skills or industry skills. Testing services are increasingly shifting to quality assurance in order to ensure business transformation. The Global software testing market is experiencing changes that are imposing the move from being product centric to customer centric approaches. It's a fact that the application testing service market is currently undergoing major disruption in industry. Organizations are all becoming "digital enterprises" that need to be connected and software is all over, and these premises come much-higher demands on speed and quality of software solutions that are currently built, and that include all process from the beginning of development. Also, the study suggests that the application testing service market is a mature market, it is greatly impacted by various and strong trends and technologies, such as agile or DevOps as we have described before and of course include software automation and intelligent automation. These services are transforming at a very fast pace and even higher than in the past — in line with the quick adoption of new technologies and the speed with which they are integrated into organizations. Clients are turning more to testing service providers to meet the increased demand for improving their own in-house testing competences, as well as meeting the more complex needs that organizations have. Yet, the past few years have seen strong growth in the application testing service market. The research and study done by the Gartner Magic Quadrant show a continuous strong growth with some bifurcation in the market in terms of consistent high growth numbers. It is a fact that most providers are growing, and the weighted average is 8.4%, but there are also two providers in the Magic Quadrant this year whose revenues have declined 1.8% and 3.5%. Of course, there are some good opportunities in the software testing marketplace, but the competition is ferocious, and providers are more challenged to maintain their status to maintain their clients. As customers' seek for improved quality standards of products and services with lowering costs, the pressure is built for vendors around driving automation, and thus improved workflow, traceability and metrics capabilities in order to move up the ladder of maturity levels of testing and contribute to QA transformation. On the other hand, all software testing providers are also in the process of reskilling their workforces from manual testers to multi-skilled testers with

[141]Magic Quadrant for Application Testing Services, Worldwide Published: 21 November 2017 ID: G00321447. https://www.tcs.com/content/dam/tcs/pdf/discover-tcs/about-us/analystreport/magic-quadrant-for-application-testing-worldwide.pdf

an understanding and competency in automation, development, technology and business process. The resulting rate of change is high among the providers in the Magic Quadrant, but the result reached so far varies from provider to provider. Figure 25 shows the key vendors. Among them there are Atos and IBM that are involved in the ElasTest project. Specifically, Atos is a good fit for clients that need strong local support or nearshore capabilities in EMEA (Europe, the Middle East and Africa) and for clients with high security demands. Gartner estimates that the year-over-year growth in Atos' application testing service revenue from the third quarter of 2016 through the second quarter of 2017 was 1.7%. That may present an opportunity to leverage the outcomes of the project and investigate more. The same will be for IBM, as involved in the ElasTest project. IBM is a Leader; it has an estimated 27,800 dedicated application testing service resources. Gartner estimates that the year-over-year growth in IBM's application testing service revenue from the third quarter of 2016 through the second quarter of 2017 was 8.8%. Over 10% of application testing service revenue comes from each of the following industries: banking, insurance and telecommunications. IBM has developed a test case optimization capability that maximizes test case coverage with an increasing reduction of number of test cases. This is supported by a dedicated tooling strategy and is aimed at supporting quality engineering client strategies and the need to minimize waste in support of agile and DevOps models. Also, 15% of IBM's revenue originates from clients with less than 999 employees, 34% from clients with 1,000 to 10,000 employees and 51% from clients with more than 10,000 employees.

Figure 25. Magic Quadrant for Application Testing Services, Worldwide

6.6 Expectations from the Market

The report World Qualiy Report by CapGemini [36], says that the respondents will increase 32% of investments in testing IT spending. It suggests to invest also more in testing to protect their brands. Also, along these findings, the same report that includes answers from 1660 CIOs (Chief Information Officers) of 32 countries acknowledges during last year 2017 quite important facts to software testing trends: as shown in Figure 26, there's a 40% say that key factors are that testing allows to "*detect software defects before delivery*", but most important that 41% agree that helps "*increase the software quality or product*". Whereas, also 34% help to ensure end-user satisfaction. Also, they suggest that quality checks are somehow tackled while performing testing activities and these are between 30% and 28% of respondent's answers. In less impact we understand that protecting the brand, or increase quality among all disciplines or reducing waste are less considered as to 26% to 21%.

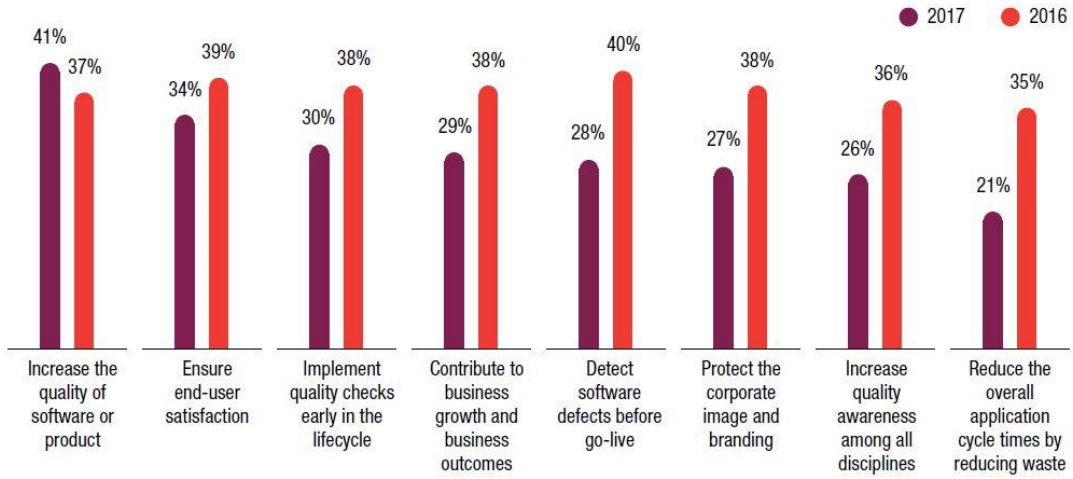

Figure 26. Reasons why testing is important in organizations

Also, "Test automation" is a very important concept, seen as the main approach to shortening the testing and delivery time. The complexities with applications and software are increasing day by day and there is a growing need for continuous software testing, which might reach a dead point if automation is not implemented. Test Automation is the tendency and of course can bring consistency and velocity to the testing activity. Major open-source and free tools such as Selenium and Katalon and commercial tools as Ranorex and TestComplete now support integration with many DevOps tool-chains like Jenkins, GIT, and JIRA. While being an enabler, it might be a challenge for testers to adopt the right strategy for test automation. It will be necessary to understand and have the right skills in test automation.

6.7 Comparison of ElasTest with trendy tools

Among the tools described in Section 4, we want to analyze in depth in this section the following key tools due to their similarity and/or relevance to ElasTest: **TestComplete, qTest Explorer, PractiTest, TestLink, HP ALM Quality Center, Jenkins, Travis CI and EasyQA.** We refer to Section 4 for technical details of these tools, here we want to focus on specific market aspects such as ease of use, business model, cost. These tools help

solving only a part of the problems covered by ElasTest. As we have said before, we consider ElasTest a cloud platform that aims to ease end-to-end testing of complex distributed systems or in the cloud.

The two key features of the platform are:

1. provide an easy deployment process and easy access to the necessary services usually involved in an end-to-end test, and,
2. provide easy-to-use tools to show and analyze logs and metrics of all elements involved in end-to-end testing activities.

ElasTest aims to improve the efficiency and effectiveness of the testing process to improve the overall quality of large software systems. All of these tools may become underlying components of the platform and could be adopted within ElasTest**.** Table 44 shows an overview of these tools compared to ElasTest. We performed this comparison in terms of: strengths and weaknesses of the tool, similarity percentage with ElasTest, deployment model, automation, type of application under test, acceptance by market/developers, numbers of users and customers, number of test, target market, ease of installation and use, finally business model and cost.

	ElasTest	TestComplete	qTest Explorer	PractiTest	TestLink	HP ALM Quality Center	Jenkins	Travis CI	EasyQA
Value Proposition (strength)	ElasTest: making complex testing simple. a Testing Platform. Improving the efficiency, productivity and code reusability of the testing process in large complex distributed applications	Gives testers the ability to create automated tests in Windows, Web & Androi and IOs. Screen capture to create scripts Easy debug scripts 1,500 mobile and desktop browsers in more than 65 operating systems	Simple and intuitive to use Easy-to-use UI and low ramp-up time. Records every step and screen of testing execution Take notes and annotate while testing. Archives complete test session recordings. qTest for Jira Test Management is the complete testing and QA platform providing test coverage and	Solution for Agile Testing, Regression Testing, MicroServices and DevOps Reports and dashboards are very customizable so to create metrics	Testlink is one of the best and usefull test tracking tools	The software quality management component of the highly renowned HP application lifecycle management (ALM) software suite	A vibrant DevOps Automation Community	Test and deployment continuous integration service	Test managemen t tool for each IT Developme nt team to improve software quality.

			bug reporting for Jira issues						
Weakness of the tool	Integration complexity	Not found	Can't call tests from other projects	Hard to integrate questions. Option to copy and paste tests and ability of batch editing tags	The design and ergonomic is old	None	None	None	None
% Similarity with ElasTest	N/A	80%	80%	70%	70%	70%	70%	70%	50%
Deployment model	Cloud, SaaS	Cloud, SaaS	Cloud, SaaS	Cloud, SaaS	OS Web-based tool	Cloud, SaaS	Cloud, SaaS	Cloud, SaaS	Cloud, SaaS, Web
Automated Test platform	Yes	Yes	Yes	Yes (SaaS)	Yes	Yes	Yes	Yes	Yes (SaaS)
Application under test	web, mobile apps	web, mobile apps	web, mobile apps	web, mobile apps	web, mobile apps	web, mobile apps	web, mobile apps	web, mobile apps	Windows desktop, web, mobile apps
Acceptance by market/developers	Low	Niche	Very High	Low	Medium	Low	High	Medium-High	Low

Statistics Of usage, numbers of users	N/A	300,000 users worldwide	No info	No info	No info	No info	>150,00 active users worldwide	>900k open source projects	Not information
Nº of test run	29 commits	>10 million tests	No info	>2.5 million tests run on customers >450K test cases	No info	No info	>148.416 installation's 1,000 plugins >6 million build jobs on Jenkins	>200K active projects	No info
Nº customers	N/A	>6 million individual developers	>500 customers	No info	More than 25 years. No info on number of users	No info	No info	>600k users	No info
Targeted Market	Developers Community Testers designers Students SMEs	Companies, SMEs, Large and medium enterprises IT professionals, Developers and QA teams	SMEs, Large and medium enterprises QA teams, developers	Technical teams, developers, public administration	Managers, QA teams, developers	Companies, SMEs, Large and medium enterprises Free users	Developers, SMEs, Large and medium enterprises	Developers, Startups, SMEs, big corporations	Companies, SMEs, Large and medium enterprises
Ease of installation and use	Mid-user friendly	Easy to setup and run. Easy to use and help desk very active	Easy to setup and run	Not really	Mid-user friendly	Easy to setup and run	Very easy to use	Travis CI is simple to use	Mid-user friendly

Business model	Open Source + free	License and maintenance fees. Pay per use	Free License +Hosting Pay per use	Pay per use + hosting	Open Source + free	Open Source + hosting License, Pay per use	Open Source + hosting	High	Pay per use
Cost	Not yet defined	$1,250 (year)	$29/user/month	$35 PROFESIONAL/user/month $45ENTERPRISE/user/month	Free	$9,000 per seat license	Free	Various prices $69/month for hobby projects $129/ month Startup $249/ month Small Business $489/ month for larger teams	$10/user/month

Table 44. ElasTest vs key tools

We defined the percentage of similarity namely the similar features of the tool with respect to ElasTest according to our perception of the market tools that can be subject to changes or discussed. For defining if a tool is user friendly, then easy to use, we have revised comments of experts reviews, and we have analyzed the level of complexity, namely if the tool is found intuitive enough by its users, if it's easy to use for beginners, etc. Below a brief description of the most important market aspects of the relevant tools:

- qTest is one of fastest growing test management solutions on the market today amongst Agile testing and development teams. Its main benefit is that qTest provides software testing and development teams with an easy to learn, easy to use, lightning fast, scalable test management that can be integrated with JIRA, ALMs, and automation tools. qTest contributes to the whole QA process making it faster, simpler and more efficient.
- TestLink is considered as one of the best and useful test tracking tools and truly open source test management application available for use in the market. It is a web-based tool and has some features like requirement management, test case creation, test runs, tracking bugs, maintenance, reports as well as seamlessly integration with common issue trackers. A lot of companies are using TestLink because it is good and it's easy to use, to create projects, add test cases, assign them to test engineers and track the test execution based on individual or project. It's very easy to maintain and track the progress in this single tool. It seems also, that developers don't need expertise in using this tool. Anybody with basic computer knowledge can start using the tool on their own without needing support or help. Also it gives the percentage of coverage per individual or project which is very essential for any manager to track the activities that are done and plan ahead software releases. For example, Testlink is used in Atos and in the industry has good acceptance among developers and tester teams.
- Jenkins is a continuous integration tool and one of the most easily configurable and easily managed tools for on-boarding applications for continuous integration and continuous delivery. It allows to configure all considering Jobs and plugins. Also, Jenkins can be integrated with different source control management tools, build automation tools and code quality tools. Similar to Testlink, it has some features which are helpful for users that have no experience, and can get started and perform activities such as integrating the source code management tool, build automation tool, deployment scripts, code quality tool, select servers to deploy applications and other available options and plugins.
- Travis CI is a very popular tool, as shown in their website it has more than 900k open source projects that are using Travis CI, it is a continuous integration platform and allows developers (currently counts with more than 600k users) to run various processes after code has been changed as well as to make tests and assure that they pass before deployment.

6.8 Market Perspective for ElasTest

The market analysis evidenced that the IT market is expected to be continuously growing. Organizations investments are progressively shifting from traditional and well-known IT offerings to cloud services.

The analysed studies highlight the rising importance of Continuous Integration, DevOps, and Agile as factors that are driving the market whereas the investments devoted to software testing for quality assurance will take 40% of the total IT budgets worldwide. Specifically, IoT testing market size is estimated to grow very fast. This clearly shows that companies do understand the importance of software testing for improving software quality.

However, the market analysis outlines that the cost of the management of test environments within companies still remains a challenging task and this implies the need of efficient testing strategies and activities. In this context, the following market opportunities are detected for the ElasTest project:

- ElasTest envisions a cloud-based platform that can be offered on the IT market in order to enable/ease the creation of a new generation of software testing services. This platform fully satisfies the IT market needs. Specifically, the technical solutions promoted by the Elastest project have been natively conceived in order to be integrated with any Agile process, Continuous Integration platforms, and DevOps practices. Moreover, some of the technologies and one of the verticals developed within the project address the testing of IoT solutions.
- ElasTest project promotes an open cloud platform for software testing. This allows the definition of reliable and reusable testing component/services, the improvement of the efficacy of the testing-based QA procedure as well as the reduction of the overall costs of software testing activities in the long term.
- ATOS and IBM, which are involved in the project, are among the key vendors and the most important providers of software testing technologies and their application testing revenues are expected to grow. This represents an opportunity to be leveraged to exploit the outcomes of ElasTest from both technology and research perspectives.

7 Conclusions

This deliverable presented the state of the art about all scientific and technical aspects that are of interest in ElasTest. The results of the performed systematic survey on cloud testing showed a growing interest of the scientific literature for innovative cloud testing solutions leveraging the efficiency and the computational power of the cloud. Test design and execution are the main addressed testing activities whereas lot attention is also devoted to evaluation of not functional aspects such as scalability, elasticity, reliability, security and so on. These results confirmed that ElasTest goes exactly towards the current research direction in cloud testing by providing a comprehensive cloud platform able to address both the design of complex tests and their execution for the validation of large software systems, particularly in what refers to non-functional

properties such as reliability, security and scalability. The main research directions of ElasTest deal with test orchestration and recommendation. The analysis of scientific literature showed that they are very new research topics of the project. Concerning the former topic, the results of a survey evidenced that the test orchestration notion as proposed in ElasTest is completely new, this is a very challenging research topic and a very few approaches are targeting it in literature. About the latter research topic, the results of a systematic survey revealed that few works target recommender systems applied to software testing phase, and these works belong to two main classes: code recommendations and test cases prioritization recommendations that are different from the recommendation notion of ElasTest aimed to support the tester during test cases design by suggesting the specific T-Job combinations to be included into a TiL.

The analysis of technical SotA raised a huge number of tools and specific solutions in the different technological areas of the project. Some of these tools have been integrated into ElasTest. This analysis evidenced at the same time the lack of a comprehensive and general platform aimed to end-to-end testing of large complex cloud applications, which could include web applications, mobile, WebRTC, and so on. ElasTest will advance SotA in all the considered SotA aspects as evidenced also in Table 42 by providing such a general platform for automating testing all along the test process cycle, including SUT deployment, test execution, SUT monitoring during test run, and reporting of test results. Finally, market analysis revealed increasing investments in testing of IT systems and the demand of competitive solutions for cloud testing. These aspects can positively impact on the development of ElasTest aiming to improve the efficiency and effectiveness of the testing process enhancing the overall quality of large software systems.

8 References

[1] ElasTest project Description of Action (DoA) – part B. Amendment 1. Reference Ares (2017)343382. 23 January 2017.

[2] J. Gao, X. Bai, and W.-T Tsai. Cloud testing-issues, challenges, needs and practice. Software Engineering: An Int. Journal, 1(1):9–23, 2011.

[3] I. Chana, A. Rana, et al. Empirical evaluation of cloud-based testing techniques: a systematic review. ACM SIGSOFT Software Engineering Notes, 37(3):1–9, 2012.

[4] J. Gao, X. Bai, W-T Tsai, and T. Uehara. Testing as a service (TaaS) on clouds. In IEEE 7th Int. Symp. on Service Oriented System Engineering (SOSE), pages 212–223. IEEE, 2013.

[5] A. Bertolino, G. De Angelis, M. Gallego, B. Garcia, F. Gortazar, F. Lonetti and E. Marchetti, A Systematic Review on Cloud Testing. Submitted for publication to journal. June 2018

[6] J. Gao, W.-T. Tsai, R. Paul, X. Bai, and T. Uehara. 2014. Mobile testing-as-a-service (MTaaS) - Infrastructures, issues, solutions and needs. In Proceedings of 15th International Symposium on High-Assurance Systems Engineering, HASE 2014. 158–167

[7] A. Ali and N. Badr. 2016. Performance testing as a service for web applications. In IEEE 7th International Conference on Intelligent Computing and Information Systems, ICICIS 2015. 356–361.

[8] C.-H. Liu, S.-L. Chen, and W.-K. Chen. Cost-benefit evaluation on parallel execution for improving test efficiency over cloud. In IEEE International Conference on Applied System Innovation (ICASI), 2017, 199–202.

[9] C. Liu and S. Chen. Evaluation of cloud testing strategies based on task decomposition and allocation for improving test efficiency. In International Conference on Applied System Innovation, IEEE ICASI 2016.

[10] G. Hwang, C. Wu-Lee, Y. Tung, C. Chuang, and S. Wu. Implementing TaaS-based stress testing by MapReduce computing model. In Proceedings of the IEEE International Conference on Software Engineering and Service Sciences, ICSESS. 2014, 137–140.

[11] S. G. Gaikwad, M.A. Shah. Pipeline Orchestration for Test Automation using Extended Buildbot Architecture. Conference on Emerging Applications of Electronics System, Signal Processing and Computing Technologies (NCESC 2015), pp. 21-25

[12] H.-H Deyab, and RB Atan. Orchestration framework for automated Ajax-based web application testing. In MySEC, 2015, pp. 1-6.

[13] Y. Falcone, JC. Fernandez, L. Mounier, JL. Richier. A compositional testing framework driven by partial specifications. InTesting of Software and Communicating Systems 2007 (pp. 107-122). Springer, Berlin, Heidelberg.

[14] N. Rathod and A. Surve. Test orchestration a framework for Continuous Integration and Continuous deployment. International Conference on Pervasive Computing, 2015

[15] B. Bergelson and I. Exman Dynamic test composition in hierarchical software testing. 24th Convention of Electrical and Electronics Engineers in Israel, 2006.

[16] H.-J. Happel and W. Maalej, Potentials and challenges of recommendation systems for software development, in Proceedings of the 2008 international workshop on Recommendation systems for software engineering, 2008, pp. 11–15

[17] M. Robillard, R. Walker, and T. Zimmermann, Recommendation Systems for Software Engineering IEEE Software, vol. 27, no. 4, pp. 80–86, Jul. 2010.

[18] U. Pakdeetrakulwong, P. Wongthongtham, and W. V. Siricharoen, Recommendation systems for software engineering: A survey from software development life cycle phase perspective, in The 9th International Conference for Internet Technology and Secured Transactions (ICITST-2014), 2014, pp. 137–142.

[19] M. Gasparic and A. Janes, What recommendation systems for software engineering recommend: A systematic literature review, Journal of Systems and Software, vol. 113, pp. 101–113, Mar. 2016.

[20] B. Kitchenham and S. Charters, Guidelines for performing Systematic Literature

Reviews in Software Engineering. 2007.

[21] Basili, V. R.,Trendowicz, A.,Kowalczyk, M.,Heidrich, J.,Seaman, C.,Münch, J.,Rombach,D., 2014. Aligning Organizations Through Measurement: The GQM Strategies Approach. The Fraunhofer IESE Series on Software and Systems Engineering. Springer International Publishing.

[22]M. Erfani, I. Keivanloo, and J. Rilling, Opportunities for Clone Detection in Test Case Recommendation, in Proceedings - International Computer Software and Applications Conference, 2013, pp. 65–70.

[23] J. Ge and J. Liu, Software test cases recommendation system research based on collaborative filtering, in 2016 IEEE/ACIS 17th International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, SNPD 2016, 2016, pp. 557–562.

[24] W. Janjic and C. Atkinson, Utilizing software reuse experience for automated test recommendation, in 8th International Workshop on Automation of Software Test (AST), 2013, pp. 100–106.

[25] S. Kpodjedo, F. Ricca, P. Galinier, and G. Antoniol, Not all classes are created equal: toward a recommendation system for focusing testing, in Proceedings of the 2008 international workshop on Recommendation systems for software engineering, 2008, pp. 6–10.

[26]B. Miranda, E. H. da S. Aranha, and J. M. Iyoda, Recommender systems for manual testing: deciding how to assign tests in a test team, in Proceedings of the ACM-IEEE international symposium on Empirical software engineering and measurement, 2012, pp. 201–210.

[27]B. Ashok, J. Joy, H. Liang, S. K. Rajamani, G. Srinivasa, and V. Vangala, DebugAdvisor: a recommender system for debugging, 2009, p. 373.

[28]N. Bettenburg, S. Just, A. Schröter, C. Weiss, R. Premraj, and T. Zimmermann, What Makes a Good Bug Report?, in Proceedings of the 16th ACM SIGSOFT International Symposium on Foundations of Software Engineering, New York, NY, USA, 2008, pp. 308–318.

[29] J. L. Rodas, D. Méndez-Acuña, J. A. Galindo, D. Benavides, and J. Cárdenas, Towards testing variability intensive systems using user reviews, in 2015 10th Computing Colombian Conference (10CCC), 2015, pp. 39–46.

[30]H.-T. Cheng et al., Wide & deep learning for recommender systems, in Proceedings of the 1st Workshop on Deep Learning for Recommender Systems, 2016, pp. 7–10.

[31] B. Hidasi, A. Karatzoglou, L. Baltrunas, and T. Domonkos, Session-based recommendations with recurrent neural networks, presented at the ICLR, 2016.

[32] H. Wang, N. Wang, and D.-Y. Yeung, Collaborative Deep Learning for Recommender Systems, 2015, pp. 1235–1244.

[33] R.P. Bush, J.F. Hair, Marketing research in a Digital International Environment, 4th ed. McGraw-Hill, 2009

[34] CapGemini –World Quality Report, 2017-2018 9th Edition. https://www.capgemini.com/service/world-quality-report-2017-18/

[35] Internet of Things (IoT) Testing Market by Testing Type (Functional, Performance, Network, Security, Compatibility, and Usability), Service Type (Professional and Managed), Application Type, and Region - Global Forecast to 2021. https://www.marketsandmarkets.com/

[36] CapGemini -World Quality Report September 14, 2016, https://www.capgemini.com/resources/world-quality-report-2016-17/

[37] B. García, L. L. Fernández, M. Gallego, and F. Gortázar. Testing Framework for WebRTC Services. In Proceedings of the 9th EAI International Conference on Mobile Multimedia Communications, pp. 40-47. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2016.

[38] ElasTest project D4.1 Test Orchestration basic toolbox v1. June 2018.

[39] ElasTest project. D2.3 ElasTest requirements use-cases and architecture v1. June 2018.

[40]Elastest project. D3.1 ElasTest Platform Cloud Modules v.1. June 2018

[41]ElasTest project D6.1: ElasTest Continuous Integration and Validation System v1 June 2018

[42]ElasTest project. D6.2 ElasTest platform toolbox and integrations. June 2018

[43]The Forrester Wave™: Continuous Testing Service Providers, Q3 2017. https://go.forrester.com/blogs/the-forrester-wave-continuous-testing-service-providers-q3-2017/

[44] ElasTest project D4.2 Test Recommendation Engines v1. June 2018

[45] F. Lonetti, E. Marchetti. Emerging Software Testing Technologies. Advances in Computers 108: 91-143 (2018)