Project N°: **FP7-610582**

Project Acronym: **ENVISAGE**

Project Title: **Engineering Virtualized Services**

Instrument: **Collaborative Project**

Scheme: **Information & Communication Technologies**

Deliverable D5.6
Market Analysis and Initial Exploitation Plan

Date of document: T15

Start date of the project: **1st October 2013** Duration: **36 months**

Organisation name of lead contractor for this deliverable: **ENG**

Final version

STREP Project supported by the 7th Framework Programme of the EC		
Dissemination level		
PU	Public	✓
PP	Restricted to other programme participants (including Commission Services)	
RE	Restricted to a group specified by the consortium (including Commission Services)	
CO	Confidential, only for members of the consortium (including Commission Services)	

Executive Summary:
Market Analysis and Initial Exploitation Plan

This document summarises deliverable D5.6 of project FP7-610582 (Envisage), a Collaborative Project supported by the 7th Framework Programme of the EC. within the Information & Communication Technologies scheme. Full information on this project is available online at `http://www.envisage-project.eu`.

D5.6 provides information on preliminary results of the market analysis and exploitation activities. It describes the exploitable items and ENVISAGE solution, it provides information on the analyses of the target market and how ENVISAGE solution addresses its needs. The deliverable presents also the initial joint exploitation plan and the exploitation intentions for each partner.

List of Authors

Domenico Presenza (ENG)

Isabel Matranga (ENG)

Giuseppa Caruso (ENG)

Reiner Hähnle (TUD)

Elvira Albert (UCM)

Amund Tveit (ATB)

David Costa (FRH)

Cosimo Laneve - (BOL)

Behrooz Nobakht (FRH)

Frank de Boer (CWI)

Einar Broch Johnsen (UIO)

Contents

Chapter 1

Introduction

In this document we present the results of the market analysis and the initial exploitation activities performed in the first year of the ENVISAGE project.Envisage develops a semantic foundation for virtualization and SLA that both goes beyond and supports today's cloud technologies. This foundation makes it possible to efficiently develop SLA-aware and scalable services, supported by highly automated analysis tools using formal methods. As shown in the figure 1.1 - Envisage tools supports development and maintenance of the three main types of cloud services:

1. SaaS - user-facing applications
2. PaaS - high-level platforms and APIs for developers
3. IaaS - low-level infrastructure.

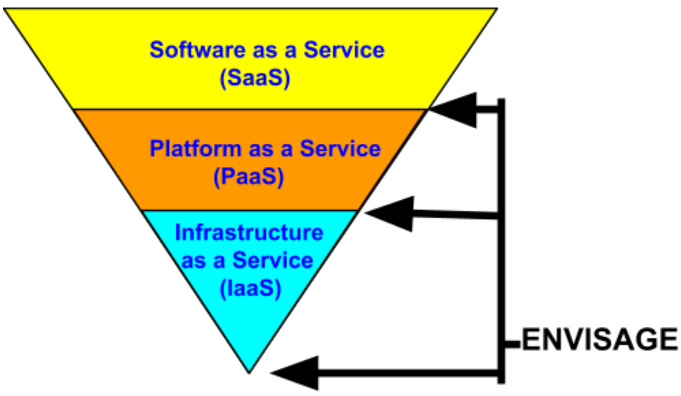


Figure 1.1: ENVISAGE vs Cloud Services

As it is shown in the market analysis chapter, the cloud services market is continually growing and becoming the default operational model adopted in the ICT infrastructure of the European industry. Cloud-based software is a fast-growing segment of the global software market. In "Sizing the Cloud," Forrester Research predicts that "'the SaaS market will increase over 600% this decade, from 40.7$ billion in 2011 to 241$ billion by 2020"'. With regard to the Paas and Iaas, the analysts predict a compound annual growth rate 30% until 2017 for the former and a growth of 37% per year for the latter.
In this period the ENVISAGE team has identified the project exploitable items and their innovative features which are presented in chapter 2 after a brief introduction of the ENVISAGE context. In chapter 3, we give an overview of the reference market for the ENVISAGE project results and describe its trends, size and growth. We also provide some information on how the project solution may address the needs of such a market. In the same chapter we identify and analyse a set of current and potential competitors and describe the added value provided by the ENVISAGE solution. In chapter 4 we present ENVISAGE value proposition

which rides on the activities and analysis performed and described in the previous chapter. In this section we look at the needs addressed by ENVISAGE solution and its potential customers. Finally in chapter 5 we present the initial exploitation plan both from the point of view of the Consortium as a whole and of each single partner. The exploitation activities reported in this document and its results need to be considered in their initial phase. Updates on the exploitation activities and results will be given in the following versions of the deliverable (D5.7 - Business Strategy & Revised Exploitation Plan and D5.8 Standardization Activities & Final Exploitation).

Chapter 2

ENVISAGE Background and Results

2.1 The Context

Companies increasingly employ the cloud computing model, which offers undeniable added value and compelling business drivers[9]. One such driver is elasticity: businesses pay for computing resources when they are needed, instead of provisioning in advance with huge upfront investments. Another is agility: new services are quickly and flexibly deployed on the market at a reduced cost. A virtualized computing environment, such as a cloud, is an execution environment with elastic resource provisioning, several stakeholders, and a metered service at multiple granularities for a specified level of quality of service [29]. A host of a virtualized system presents a number of services to client applications, including infrastructure and platform functionalities and software services for virtualizing deployment of resources. This virtualization provides an elastic amount of resources to application-level services, thus making it possible, for example, to allocate a changing processing capacity to a service according to the demand. The figure below (from top to bottom) illustrates the end-user layer, the application-level services layer, and the resource level services layer of a virtualized system.

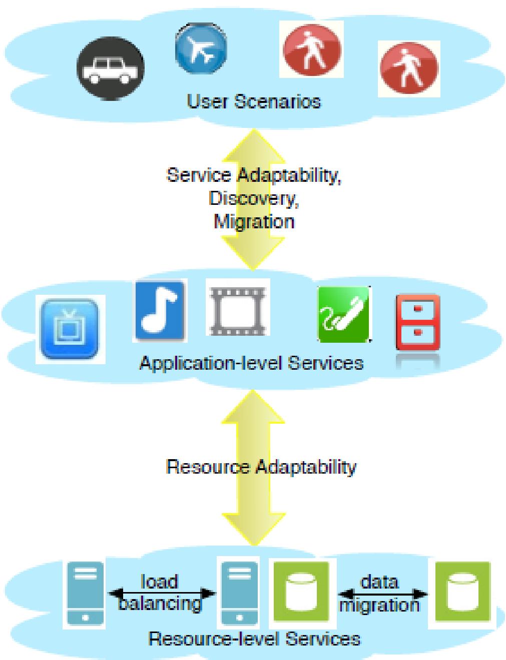


Figure 2.1: Virtualized system layers

In virtualized systems, resource provisioning is regulated by a legal contract between the service owner and the virtualized system provider, called: Service-Level Agreement (SLA). SLAs provide the fundamental ground for the interactions which take place in the above scenario and more broadly in a cloud ecosystem [27]. The SLAs sets: (i) goals through Quality of Service (QoS) attributes, (ii) privacy and protection constraints through Quality of Protection (QoP) attributes, (iii) expectations through the description of actions that need to be taken in order to deliver the service according to the QoS attributes, (iv) responsibilities through the inclusion of obligations of parties including penalties and exclusion terms, (v) evolvement cases through the definition of rules that enable efficient adaptation of resource provisioning based on the dynamic demands of the applications and the end users. By analysing the state of the art in the domain of SLAs two interesting points arise:

- the SLAs are not integrated in the programming models, which severely restricts the developer in fine-tuning the service to the available virtualized resources
- Currently, resources and resource management first appear in the service engineering process only at deployment phase, which makes it difficult and costly to fine-tune complex services to the expected runtime cost and quality of service.

To fully exploit the potential of virtualization, to make services both scalable and cost efficient, both deployment parameters and SLA compliance need to become part of the software design phase. The current modeling and analysis techniques make it highly problematic for the software developer to realistically predict the resource requirements of the developed service at an early design stage. Languages and tools for application development lack high-level support to systematically analyze performance under different resource assumptions and to compare different resource management policies. Furthermore, services for virtualized environments require descriptions of resource-dependent and resource-aware behaviors that are based on abstract yet precise execution models. This would enable a better exploitation of runtime resources, as well as lower development costs and shorter time to market for service developers. ENVISAGE develops a formal foundation and practical tool for the modeling, analysis, and verification of application level SLA-aware services in virtualized environments, with focus on model-based analysis of quantitative (or non-functional) aspects of SLAs. ENVISAGE offers a development framework for software-as-a-service in a virtualized environment which can be integrated into existing industrial software development processes. The framework consists of an integrated suite of advanced methods and tools to engineer complex services and that enables:

- a design-by-contract methodology including service-level agreements,
- the definition of application-level services with resource requirements
- the modeling of deployment scenarios reflecting elastic, virtualized architectures monitoring capabilities assuring adaptability to failures and to renegotiations of service-level agreements
- the systematic analysis of quality of service behaviors at early stages in software development.

The framework is realized by:

- including **formal methods** into the software engineering practice, which are used to analyze resource needs at the design stage and to generate autonomous monitors of service-level agreements to address failures after deployment.
- developing a **novel abstract behavioral specification languag**e to leverage resources and their dynamic management to the abstraction level of models. The abstract behavioral specification language will stay close to a high-level programming language to be easily usable and accessible to software developers.

2.1.1 ABS modelling language

Envisage is based on the modelling language ABS (Abstract Behavioral Specification). Here we briefly describe the ABS eco system and the development prospects of ABS. The ABS language is the main outcome of the FPT FET Integrated Project - Highly Adaptable and Trustworthy Software using Formal Models (HATS[1]) - that ran from April 2009 until March 2013. The ABS eco system consists of two main parts: the ABS modelling language and the ABS tool set which we describe now briefly in turn.

The ABS Modelling Language

The ABS modelling language is a fully executable, concurrent language whose basis is a modern object-oriented/functional design paradigm. ABS has a formal semantics and is extensively documented in [1]. There is also a tutorial on ABS [19]. Compared to other modelling languages, ABS has a number of distinctive features:

- ABS permits automatic code generation. Currently supported target languages include Java, Erlang, and Haskell.
- ABS includes sublanguages for feature description and to model product variability. Specifically, it is possible to define product lines in ABS [20]. Together with code generation, this makes end-to-end modelling of product lines possible.
- ABS can reflect the properties of its runtime environment. Deployment components allow to access and process resource parameters of the machine(s) where it is running. This is crucial for its intended usage within ENVISAGE.
- ABS has been developed in tandem with a number of state-of-art software productivity tools to assure their scalability (see next section).

ABS is well established in the academic formal methods and programming languages communities. For example, the standard ABS reference [26] received well over 100 citations in Google Scholar in the three years since its publication. Tutorials about ABS have been presented at major international conferences, such as ECOOP. A summer school [8] and a track of an international conference [21] were to a large extent devoted to ABS. A dedicated website for ABS exists that is continuously developed.

The ABS Tool Set

ABS comes with a number of mature and sophisticated tools [31] that go well beyond what is available in commercialIDEs. The ABS tool set contains simulators, type checkers, code generators (for Java, Haskell, Erlang), glass box test generation, complete deadlock and livelock analysis, resource analysis, formal verification. Several of them are described in the tutorial [10]. The ENVISAGE tool set is provided both as a command line interface and as an Eclipse extension. The latter provides an Eclipse-based IDE for ABS that is similar to the Java Eclipse IDE. It gives access to most ABS tools as extensions of the standard Eclipse context menus so that users need no expert knowledge to make us of, for example, deadlock or resource analysis. Most of the ABS tools will be extended to cover the new ABS language and specification elements to be added during ENVISAGE's Work Tasks 1.2 and 1.3. Having ABS and the ABS tool set as a starting point makes the ambitious tool coverage of ENVISAGE possible in the first place.

[1] project-home page http://www.hats-project.eu

2.1.2 The Exploitable Items

The ENVISAGE project is developing a set of innovative highly automated analysis tools using formal methods, that will support and enable to efficiently develop SLA-aware and scalable cloud services. ENVISAGE develops a formal foundation and practical tool for the modeling, analysis, and verification of application level SLA-aware services in virtualized environments. This paragraph provides a current overview of the main exploitable items developed in the project and explains the innovative features offered by each of them.

Exploitable Item name	**Description**	**Innovative Features**
Simulation Tool	The ENVISAGE Simulation Tool is a simulation environment for the abstract modeling language. It combines system level descriptions with resource and deployment models to allow rapid prototyping. The simulation environment will allow the rapid prototyping of models in different deployment scenarios and with different load balancing strategies. ENVISAGE simulation environment will integrate state of the art tools for development of formal languages and execution platforms for operational semantics with compiler generation and type checking in an IDE with editing and visualization support (such as Eclipse).	Modelling and Simulation of big data jobs (Mapreduce and Spark) to estimate resource use in advance (time and cost), and also provide recommendation of which cloud configuration (e.g. Hadoop cluster machine types) to use for cost-efficient result. The innovation is this simulation combined with utilization of all software quality related features in ABS language and supporting tools (e.g. unit test creation, static analysis for resource analysis and concurrency) to ensure improved software quality of big data jobs prior to their run through simulation (e.g. to reduce risk of jobs failing during costly runs).

Deductive Verification Framework	The ENVISAGE Deductive Verification Framework will extend KeY-ABS, a deductive verification system for ABS. KeY-ABS is based on the verification system KeY which verifies sequential Java programs and a variant of dynamic logic called ABS Dynamic Logic. KeY-ABS deals with concurrent ABS programs and uses history-based functional specification of ABS models in terms of method contracts, class invariants and interface invariants. With the Deductive Verification Framework we intend to extend the current history-based specification system to specify SLAs in terms of the restriction between resource consumption, resource provision and load balancing. Using the developed framework it will be possible to formally prove that a system modelled in ABS is able to guarantee certain resource and load balancing properties that are part of an SLA. The used logic and verification method is compositional and modular. This means it does not make any additional assumptions on the number of objects, cogs or similar (except those specifically mentioned as part of an SLA).	The main innovation is a compositional, formal verification system for ABS models that allows to reason about guarantees stated in an SLA and involves properties related to resources and load balancing.
Test Case Generation Framework	The ENVISAGE Test Case Generation Framework will provide a set of test cases which guarantee that the selected coverage criterion is achieved. It adapts aPET test case generation framework to ENVISAGE needs. The framework receives as input an ABS program, a selection of methods to be tested, and a set of parameters that include a selection of a coverage criterion. More in detail aPET will be integrated within the ABS Eclipse IDE, and also within the ENVISAGE virtual collaboratory. The generated test cases will be displayed in textual mode and, besides, it will be possible to generate automatically ABS Unit test cases. The information yield by aPET can be relevant to spot bugs during program development and also to perform regression testing.	The main innovation is that the system makes use of new strategies and heuristics for pruning redundant state exploration and also for guiding the exploration towards specific interleavings and, specifically, those that produce deadlocks.

Resource Analysis Framework	The ENVISAGE resource analysis framework allows us to determine at an early stage of software development the resource consumption of abstract behavioral specification models (ABS). The information gathered focuses on concrete components of the system (like number of executed instructions or memory consumption at each location) but also on system-level properties (like the number of spawned tasks or the amount of data transmitted among the locations of a virtual system). This will allow the users to anticipate potential bottlenecks in the locations of the system and to optimally distribute the load of work in order to fulfil the service contracts of the components services. Developers can use the resource analysis framework to check that the resource consumption of each component fulfils the service contracts, and system designers can use the framework's results to better deploy the system and avoid potential bottlenecks.	The main innovation of the framework is that it takes into account the system level and the deployment descriptions contained in the models. Thus, the framework obtains more precise results for each component and also information involving several components like data transmission and spawned tasks.
Collaboratory Methodology	The ENVISAGE project will provide a Virtual Collaboratory, a place where tools developed in the context of ENVISAGE are made available. To facilitate the integration of tools for cloud developers, in the Virtual Collaboratory, the ENVISAGE team is developing the Collaboratory Methodology in which tools output their results in some predefined text-based language which is then interpreted by several interfaces, also developed within ENVISAGE. The technology developed will be used by programmers in the ENVISAGE project to easily integrate their tools in the Virtual Collaboratory. The Virtual Collaboratory will be then available for internal and external users in order to easily experiment with the developed tools.	The main innovation of the Virtual Collaboratory methodology is that it supports developers to rapidly develop user interfaces for analysis tools. The methodology is based on providing (1) a text-based GUI language for specifying the output, and (2) a set of generic interfaces that are able to interpret this language and view it graphically in several development environment (e.g., web-interface). Following this methodology, the Virtual Collaboratory will include such text-based GUI language and corresponding generic interface (e.g., web-interface and eclipse plugin).

Virtual Collaboratory	The Virtual Collaboratory is a (virtual) place where the tools developed in the context of ENVISAGE are made available to the community. Users will be able to experiment ENVISAGE tools but also use them in different forms, e.g., as services, through a web-interface (that is programmed in Java script), by downloading to use locally (for this we will provide an Eclipse plug in), etc. The collaboratory will also allow users to share their experience and provide feedback. It will offer an easy way for programmers to integrate their own tools (through the easy-interface language) giving the community members the opportunity to actively contribute to the ENVISAGE community.	Thanks to the Collaboratory Methodology, the Virtual Collaboratory will include text-based GUI language and corresponding generic interface (e.g., web-interface and eclipse plugin) to support developers in rapidly developing user interfaces for analysis tools. Further, there is no need to download the collaboratory since it can be used locally if they do not want to install it and, instead, it can be used through a web interface or service.

Monitoring add-ons and visualisation	**A generic model for data point expressions:** Develop a formal method approach to generalize how data points in a monitoring system can be expressed. A generic implementation model/API is delivered to enable definition, measurement, exposure and integration of data points. For example, a data point can be defined to extract QPS (query per second) from the application logs and expose it as a data point in the monitoring system using the monitoring API. Or, another data point can be defined to measure and expose the data characteristics of a customer such data categories or data API usage. **A systematic process and methodology to generate monitoring system components and plugins:** 1. Develop a methodology that enables a business stakeholder to initiate a model from a customer's contract and capture the requirements with the operations of the monitoring system. 2. Automate a process in which a customer's contract can be validated and then used to generate a model. From the model, parts of the monitoring system are generated that contribute to maintaining QoS of the service based on the customer's contract. 3. Evolve the service as necessary based on the monitoring measurements over the data points and metrics defined. When a customer contract is being negotiated, the model above allows to validate the contract against the operational aspects of the offered service. For example, if the customer's contract has a notion of QPS as a QoS metric, the model can be used to validate the offering and make tuning if necessary. In the end, the contract terms are transformed into the model and then monitoring system component for the contract are generated to monitor the service for the customer and ensure the QoS if necessary.	• An abstract formal method in software engineering is applied to capture and model the QoS metrics • The formal approach has this unique property that it is independent of any SLA specification language. • Any SLA spec or definition can be transformed using model transformation to the formal model • The formal model is implemented as an API library in the context of the project on top of ABS model and its API in Java • The API library is used to model and implemented the QoS metrics in the context of T4.3.

	Enable diverse personas to use metrics and KPIs: A diverse range of personas in a business organization demand different outcomes of a monitoring system. The monitoring data model allows a persona to define the data points as expressions in the monitoring system from their interested point of view. The monitoring system facilitates the persona to visualize the measurement of the defined data points. Thus, it helps to make an understanding of the service according to the contract. This helps the persona to build a metric or a KPI that allows business decisions. In the above example, one can build a notion of QoS over the measured QPS data point defined in the monitoring system. This can be a basis to define capacity/availability of the service that is documented in the customer's contract. Code Generator	
Code Generator	For the generation of production code from ABS models two backends will be developed: One backend for generating Java code and one for generating Haskell code.	These backends will advance the state of the art of current ABS backends with respect to performance so that ABS can also be used as an efficient programming language which can compete with for example SCALA. Added value of the ABS programming language is that it supports (1) distributed cloud applications and (2) model-based development and analysis.

Table 2.1: ENVISAGE Exploitable Items

Chapter 3

Market Analysis

This chapter aims at providing an overview of the reference market in terms of trends, size and growth for the ENVISAGE project results as well as at analysing how the project solution addresses the current market trends and needs. It provides also a gap analysis identifying the current and potential competitors and the added value provided by the ENVISAGE solution.

3.1 Market Overview

Technology has changed the way we operate, making the speed at which we do things a lot faster. Today, we operate in a global economy at Internet speed. The explosion of wireless and edge technology has raised consumer expectations. Final consumers are more informed, educated, and knowledgeable about products and services than ever before. Consequently consumers require more and more products and services with high levels of quality. These changes in the IT landscape is giving immense pressure to application service providers. Modernizing IT systems is a necessity to keep pace with the speed of change. Large, monolithic applications are giving way to smaller, more agile application components that are assembled to create composite applications to support today's demanding business processes. In this landscape, the cloud computing represents a solution for replying to those new markets needs and trends.

3.1.1 Cloud Computing Market Overview

Cloud computing is rapidly becoming the default operational model adopted in the ICT infrastructure of the European industry, due to an undeniable added value and compelling business drivers. It has been revolutionary for the IT world, with various IaaS, PaaS, SaaS and other XaaS offerings. A report from Emerson reveals that 68% of responders believe at least 60% of computing will be cloud-based by 2025 [13]. A recent Eurostat study [17] reveals that 9% of EU enterprises used cloud computing in 2014, mostly for hosting their e-mail systems and storing files in electronic form. While 46% of those firms used advanced cloud services relating to financial and accounting software applications, customer relationship management or to the use of computing power to run business applications. The highest shares of enterprises using the cloud in 2014 were observed in Finland (51%), Italy (40%), Sweden (39%) and Denmark (38%). As the cloud computing trend is spreading around the world and various application areas, cloud computing is expected to attract high business investments in the near future, as underscored by Jagdish Rebello, Ph.D., senior director and principal analyst for the cloud and big data at IHS[24]. What Dr Rebello implies is the need for increased cloud investments, indicating appropriate directions.

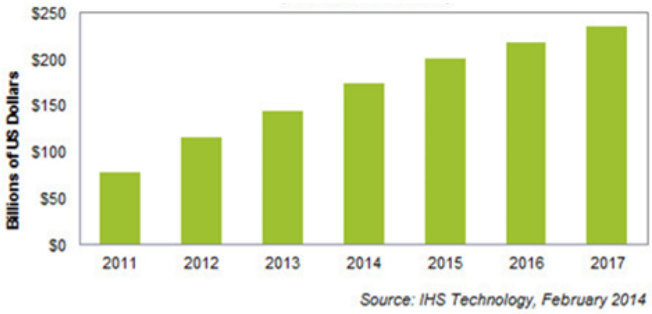


Figure 3.1: Business investments on cloud computing worldwide, IHS

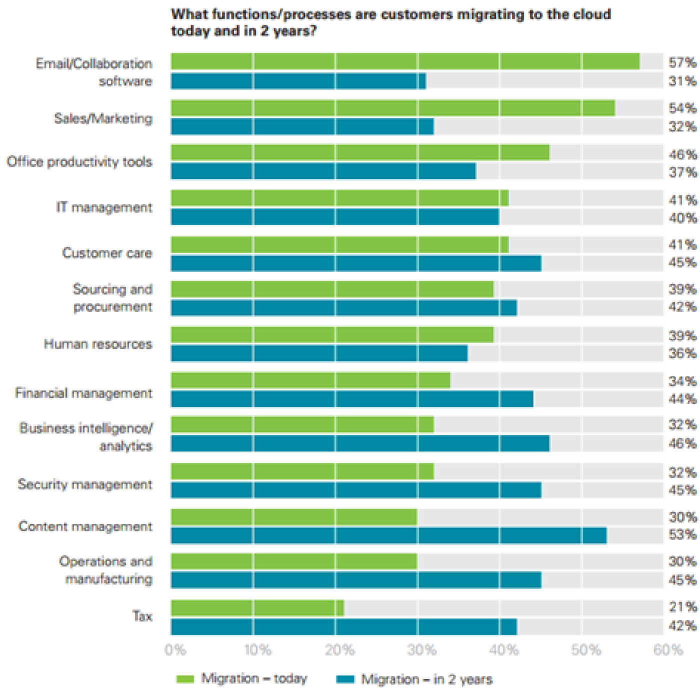
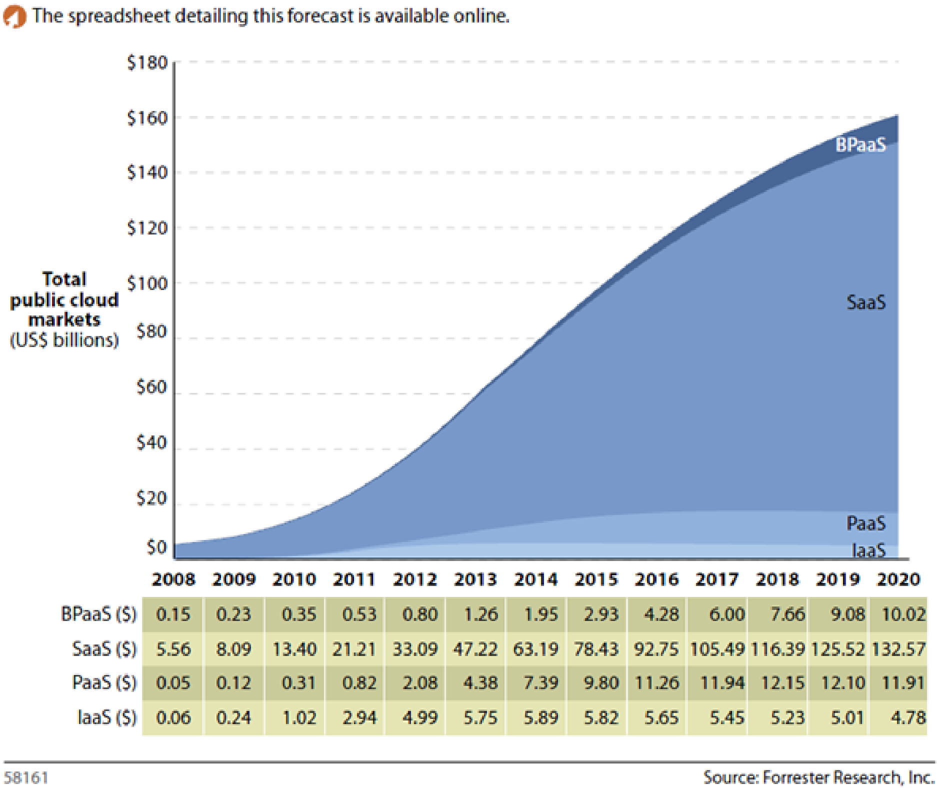


Figure 3.2: Functions migrated to cloud - Source: KPMG International's 2012 Global Cloud Providers Survey

Specifically, IHS technology analysis predicts that the investments made by businesses on cloud computing will triple within the time frame from 2011 to 2017, reaching 235.1$ billion, while the estimate for 2014 is 174.2$ billion, as shown in Figure 3.1. According to Centaur Partners, SaaS and cloud-based business application services revenue will grow from 13.5$ Billion in 2011 to 32.8$ Billion in 2016, at a CAGR of 19.5%. Moreover, Forrester Research predicts that more than 50% of enterprises will focus on developing private clouds through commercially available software until March 2015. High investments in cloud computing are also projected by Computerworld 2014 State of the Enterprise survey [16] with 42% of responders claiming to increase significantly their investments on cloud computing. IDC predictions presented in a Cisco infographic [12] show that the cloud software market will amount more than 75 billion USD by 2017 with a 5-year CAGR of 22%, while current organizations having adopted cloud solutions are expected to invest 53.7% of their IT budget in cloud solutions within the next 24 months. Finally, Accenture [2] classifies cloud computing among the top three technologies with the greatest potential for positive Return on Investment (ROI); analytics and Big Data are in the first place with 65%, cloud computing is second with a rate of 48% and mobility is classified third with 43%. Cloud computing is used by different applications today. The figure 3.2 below provides an idea about the functions that in these years are migrating to cloud according to a KPMG survey [25]. The functions that have mainly switched are email/collaboration software, sales/marketing and office productivity tools. As a key engine driving growth, sales and marketing is high on the list of functions adopting cloud today, as businesses seek innovative ways to connect with customers and manage relationships better. According this survey, the speed and agility of cloud are particularly appropriate in such cases, especially when combined with mobile devices enabling sales people to access vital data quickly.



	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020
BPaaS ($)	0.15	0.23	0.35	0.53	0.80	1.26	1.95	2.93	4.28	6.00	7.66	9.08	10.02
SaaS ($)	5.56	8.09	13.40	21.21	33.09	47.22	63.19	78.43	92.75	105.49	116.39	125.52	132.57
PaaS ($)	0.05	0.12	0.31	0.82	2.08	4.38	7.39	9.80	11.26	11.94	12.15	12.10	11.91
IaaS ($)	0.06	0.24	1.02	2.94	4.99	5.75	5.89	5.82	5.65	5.45	5.23	5.01	4.78

Figure 3.3: Global public Cloud market size 2011 to 2020, source Forrester Research

Moreover, cloud computing offers a cost-effective way to support *big data technologies*. Big Data requires huge amounts of storage space. Data storage using cloud computing is a viable option for small to medium sized businesses. In this perspective, three major reasons in particular for small to medium sized businesses to use cloud computing for big data technology implementation are hardware cost reduction, processing cost reduction, and ability to test the value of big data. Anyway the major concerns regarding cloud computing are security and loss of control.
In the next part we provide more market detail about some cloud computing market segments, such as SaaS, Iaas and Paas.

- **Software as a services (SaaS) market segment**

Cloud-based software is a fast-growing segment of global software market. According to Forrester Research in "Sizing the Cloud," the SaaS market will increase over 600% this decade, from 40.7$ billion in 2011 to 241$ billion by 2020. It is expected that Software as a service will be adopted by companies of all sizes. For this reason the development of SaaS is foreseen to be faster than PaaS, IaaS and BPaaS as it is shown in the figure 3.3 below. It is important to note that the cloud software market reached 39.3$ billion in revenue in 2013. According IDC report[14], cloud software will grow to surpass 100$ billion by 2018 at a compound annual growth rate (CAGR) of 21.3%. It is moreover expected that SaaS delivery will significantly outpace traditional software product delivery, growing nearly five times faster than the software market as a whole and becoming a significant growth driver to all functional software markets.

In this direction, Forrester's Forrsights Software Survey, 4th quarter of 2013, North American and European enterprise software decision-makers indicated that over the next two years, budgeted enterprise application spending will remain mostly for on-premises systems (53%) and licensed software that is externally hosted in a dedicated environment (25%). SaaS spending accounts for the remaining 23%. However, when considering the prior year's survey results, we see that there is movement away from on-premises systems and toward SaaS. The budget for on-premises applications is about 16% lower in 2013 than in 2012 (63% versus 53% in 2013), and SaaS spending budget is about 53% higher in 2013 than 2012 (15%, versus 23% in Q4 2013), with hosted spending holding fairly steady.
In order to analyse the SaaS industry, we use the Porter's Five Forces model (Figure 3.4). This model helps us describe a specific industry according to five driving forces, such as: rivalry among existing competitors, bargaining power of suppliers, bargaining power of buyers, threat of substitute products or services and threat of new entrants:

Figure 3.4: Porter's Five Forces

Rivalry among existing competitors
The SaaS market is characterized from a large number of firms. Consequently it is an high competitive market. Currently though their service offerings differ depending on combination and features related to IaaS, PaaS, the competition in this industry tends to stay high because product differentiation is low. According to Gartner's Forrsights Software Survey from Q4 2013, the key criteria clients use to select a SaaS solution are usually security, comprehensive service level agreements of the vendors, and compliance with local privacy laws. In this perspective, a SaaS provider that can offer a comprehensive service level agreement has a 38% more chance of being chosen.

Bargaining power of suppliers
The main SaaS suppliers are the cloud service providers (PaaS and IaaS providers). In SaaS market, considering the presence of many players the suppliers bargaining power should be low. Anyway in the last years the cloud providers have found different lock-in mechanisms in order to reduce the customers provider changes.In particular in PaaS market vendor lock-in leads to high bargaining power of suppliers, while in IaaS, brand loyalty is the main way providers can gain some bargaining power.

Bargaining power of buyers
Low switching cost and availability of many choices makes the bargaining power of buyers high in the SaaS. It is also important to say that the SaaS consumers set the standards in the industry and partly drive the aggregation of players through more demanding requirements.

Threat of substitute products or services
Threat of substitute has been low because the cloud computing technology itself is in the growth phase and SMEs tend to prefer cloud computing due to the availability of consulting services from the providers and

not existent switching cost.

Threat of new entrants
In the SaaS market, the threat of new entrants is high because in general it requires low initial investment, low time to market. Moreover the exit barriers are low. Anyway, research conducted by academia and industry in this area and enriched collaboration between academic and corporate, research and development will lead to new innovations that have the potential to bring differentiate products and services, for new entrants in the future.

- **Platform as a service (Paas) and Infrastructure-as-a-Service (IaaS) market**

Although SaaS is still the big growth area within the cloud segment, platform as a service (PaaS) is the next big up-and-comer. As more enterprises and developers move their development workloads to the cloud to take advantage of economics and flexibility inherent in as-a-service offerings, PaaS is growing in popularity. The global market for PaaS is set to reach more than 14$ billion in 2017 this is due, according to released research [15] from IDC, to the needs by companies to cut infrastructure costs and speed up application development. The compound annual growth rate for PaaS during this period will be roughly 30%, compared to the 4% growth rate this year for IT spending overall. Gartner predicts that Infrastructure-as-a-Service (IaaS) will achieve a compound annual growth rate (CAGR) of 41.3% through 2016, the fastest growing area of public cloud computing that the research firm tracks. The following graphic provides insights into relative market size by each public cloud services market segment:

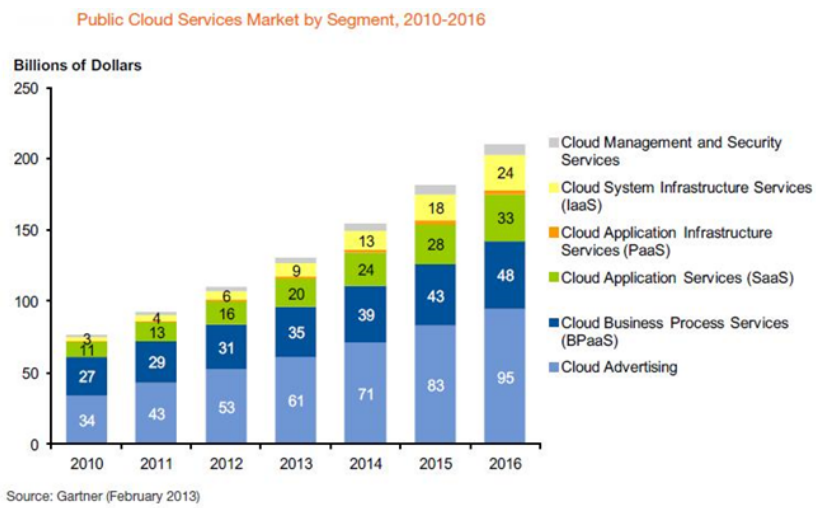


Figure 3.5: Public Cloud Services Market by segment, 2010-2016

Another recent report published by 451 Research expected rise in PaaS spending at a CAGR of 36% per year. The market is expected to exceed 20$ billion by 2016. In 2012, more than half the income of public cloud services accounted for IaaS. In this segment, analysts predict a growth of 37% per year. PaaS is the fastest growing sector, with an increase of 41 percent. It is expected by this report that PaaS will generate more than 24% of total cloud revenues. IaaS service will registered a 37% CAGR through 2016, generating 51% of cloud revenue. Other research firms have much higher forecasts for cloud computing.
Also for this industry, we use the Porter's Five Forces model in order to analyse its competitive landscape:

Rivalry among existing competitors
This market is characterized by a high number of firms and consequently by a high competitive market. Presence of almost all ICT industry such as IBM, Apple and Google is also an indication of high existing

rivalry among competitors. Amazon is the biggest name in the IaaS market. The IaaS market is also evolving quickly among leading providers, with the addition of full platform as a service (PaaS) capabilities, or at least PaaS-like capabilities into their offerings (database as a service). Consequently the Iaas vendors offer also PaaS solutions. Amazon's Elastic Beanstalk, for example, constitutes a PaaS offering from the largest IaaS vendor. According to Gartner[18], the marketplace for PaaS is a diverse and complex landscape with over 160 offerings in 15 different unique service categories provided by over 120 different vendors (some specialists and some generalists)

Bargaining power of suppliers
The Iaas and PaaS suppliers are the hardware and software vendors, but also the utilities providers. The bargaining power of suppliers usually tend to stay low.

Bargaining power of buyers
In PaaS market, lack of interoperability standards and usage of proprietary development language lead to a vendor lock-in which in turn leads to high switching cost for buyers. Thus in PaaS market, bargaining power of buyers tends to stay low due to high switching costs.

Threat of substitute products or services
IaaS is considered to be a commodity. Thus product differentiation is low and bargaining power of buyers is high. Amazon EC2 has been the early mover in the sector and is preferred by buyers mainly because of its brand name. Thus creating brand loyalty is one way to stay competitive in this market.

Threat of new entrants
In PaaS market presence of industry giants such as Google and Amazon creates a high barrier to entry which can be reduced by product differentiation and innovation. In IaaS market, threat of new entrants has been low because of the requirement of high initial investment and presence of high risk.

3.2 Market Trends vs ENVISAGE results

The market will begin to focus more on applications in the cloud rather than infrastructure. This means the migration of enterprise applications as well as new development. Thus, both the PaaS and IaaS spaces will continue to expand to support this focus on cloud-based applications, as well as processes and tools to support this migration. In most instances, these applications will operate within multi-cloud environments, which will be a mix of private clouds and public cloud providers.
Cloud management platforms will continue to dominate as the hot technologies as most enterprises move to complex multi-cloud deployments.
One of the biggest concerns users have with public cloud resources is not knowing how much they will cost, given the pay-as-you-go model. "IT shops are becoming cost centres for service delivery," says William Fellows, a researcher at the 451 Group. "But they're looking for ways to determine how their clouds are running, how much it's costing and whether it's a good value."
Moreover today another barrier to cloud computing industrial adoption are the reliability and control of resources, such as the number and kind of processors, the amount of memory and storage capacity, and the bandwidth.
Cloud performance management technologies will come into focus as enterprises understand that they will get big cloud bills and need to monitor and manage performance of these clouds to find the value. Cloud financial systems will become more relevant. They include systems that monitor clouds for use-based accounting, financial projections and so on. It becomes important as we learn to live with clouds, and thus need to monitor and allocate costs within the enterprise.
The rising trends for cloud computing adoption and investments are breeding grounds for

ENVISAGE which targets to provide a set of methods and tools for the development of applications for virtualised environments.
ENVISAGE project will offer innovative approaches for facing weaknesses related to the applications development oriented to SLA and resource issues for cloud computing as well as some relevant aspects related to software quality developments.
To make full usage of the opportunities of cloud computing, software development for the cloud demands have to address the design methodology that: (a) takes into account deployment modelling at early design stages and (b) permits the detection of *deployment errors* early and efficiently. ENVISAGE solution does it, thanks to the tools it provides, such as simulators, test generators, and static analyzers.
In this perspective, ENVISAGE will allow to better manage cost of application which run on cloud environments mainly through two ways: 1) addressing at design level SLA and resource issues; 2) improving the software quality.
Traditionally, deployment is considered a late activity during software development. The reduction of software defects at design level may reduce the organisational waste and costs. In fact, according the consolidate software architecture theory, the cost of defects can be measured by the impact of the defects and when we find them. Usually, earlier the defect is found lower is the cost of defect. If the error is made and the consequent defect is detected in the requirements or design phase, then it is relatively cheap to fix, correct and reissue it.

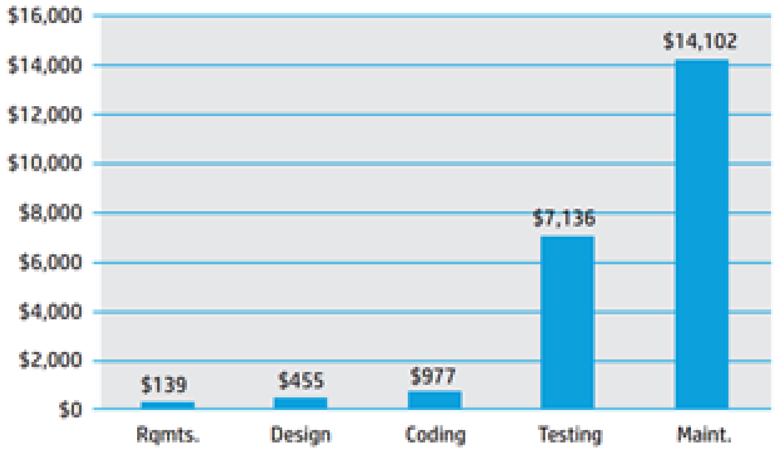


Figure 3.6: The costs of correcting defects throughout the software lifecycle

Figure 3.6 [7] shows how the cost to correct defects increases exponentially the later it is in the application lifecycle that defects are discovered. It is 50 to 100 times more costly to correct defects in testing and maintenance than correcting them in requirements and design phase. Traditional approaches to software development contain a lot of waste due to software defects. Most defects are actually introduced at the very outset of a project when business requirements are collected. More than 60% of software defects occur during this time. These defects typically go undiscovered until the testing phase of a project, and in many cases, escape through to production. The majority are not discovered until user acceptance testing, when business users realize that assumptions made about their requirements, and the interpretation of these into code, are incorrect. This cost estimation does not even consider the impact of cost due to, for example, delayed time to market, lost revenue, lost customers, and bad public relations. In the realm of cloud computing, this can be fatal considering considering the high additional complexity of resource management in virtualized environments. In such context, it is reasonable to expect even more significant differences;

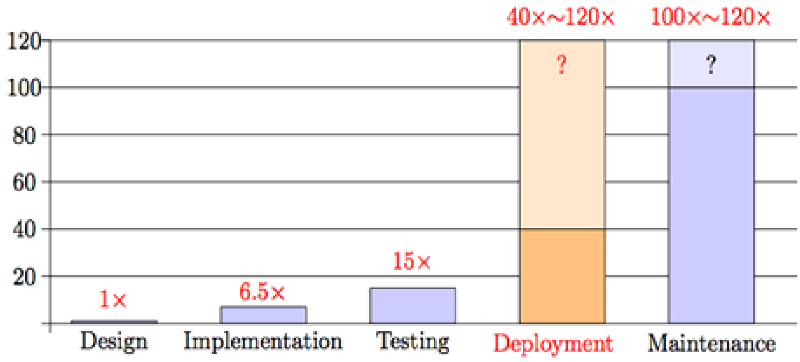


Figure 3.7: : Estimate of relative costs to fix software defects for virtualized systems with elasticity

Fig.3.7 [6] conservatively suggests ratios for virtualized software in an elastic environment. This consideration makes it clear that it is essential to detect and fix deployment errors, for example, failure to meet a service level agreement (SLA), already *in the design phase.* ENVISAGE aims at meeting these needs *by providing abstraction to reduce complexity and to obtain more compact, more understandable designs, and by supporting the automated analysis of deployment decisions in the service design phase.* In this way, the project aims to provide an answer to industry needs which require faster ways to develop software while maintaining or even improving its quality.

3.3 Gap Analysis

This paragraph provides an analysis about the current and potential competitors, identifying the added value offered by the ENVISAGE results.

3.3.1 Competitors

Large software companies offering cloud-based distributed systems rely on the correctness of a constantly increasing set of algorithms for, e.g., replication, concurrency control, fault tolerance, and scaling. *"Achieving correctness in these areas is a major engineering challenge as these algorithms interact in complex ways in order to achieve high-availability on cost-efficient infrastructure whilst also coping with relentless business-growth"*, says Chris Newcombe [11]. The added value of formal methods in this domain is the ability to handle complexity in concurrent and distributed systems and verify interactions between algorithms. For example, Amazon has used **TLA+**[1] to verify the correctness of more than 10 large and complex real-world systems.

In the context of Envisage and in particular ABS, TLA+ as a modelling language offers formal method similar in that it develops reasoning techniques based on the formal semantics of a modeling language.

Market requirements include formal methods which are "easy to understand" by being close to programming languages and offering a single framework for various analysis. Furthermore, proven technologies are desirable, i.e., with prior industrial applications and success stories for complex algorithms.

The concrete technology provided by TLA+ which has been used by Amazon consists of explicit state model checking of safety invariant using histories and of deadlock analysis. Amazon did not make use of temporal operators in their specifications. Furthermore, Amazon states that they would welcome further progress in methods for modelling and analyzing performance, improved tools for checking that executable code meets its high-level specification, and in the scalability of proof support. In Envisage, several techniques are developed to address these challenges, namely the exploitable items "Deductive Verification Framework", "Test Case generation", and "Resource Analysis Framework".

[1] http://research.microsoft.com/en-us/um/people/lamport/tla/tla.html

Considering ABS as a modeling language, it has high potential for describing Cloud-based enterprise architectures. In this regard a competitor is **ArchiMate** [2](Opedn Group Standard): "*ArchiMate offers a common language for describing the construction and operation of business processes, organizational structures, information flows, IT systems, and technical infrastructure. This insight helps stakeholders to design, assess, and communicate the consequences of decisions and changes within and between these business domains.*"
The ArchiMate modeling language distinguishes three distinct layers for describing the business processes, the supporting applications, and the computing infrastructure (e.g., hardware). Such a distinction is needed to describe the complexity of Enterprise architectures at large and their intricate interactions. New challenges to enterprise architecture languages are posed by the advent of the Cloud as an alternative computing infrastructure. A potential complementary use of the ABS is the focus on the analysis of SLA compliance based on a more detailed behavioral description of the Cloud deployment of the applications supporting the businesses.
In particular, the following features of the ABS are highly relevant in this context:

- Advantages of the use of ABS for modeling enterprise architecture, specifically modeling the Cloud deployment of the applications supporting the businesses;
- the basic object-oriented modeling concepts which smoothly integrate architecture (e.g., distribution, resources) and parallel behavior;
- simulation and rigorous (cost) analysis;
- automated monitoring, test-case generation;

3.3.2 Potential Competitors

The finalised and ongoing research projects may be considered as potential ENVISAGE's competitors.
In this perspective, the most relevante research projects related to ENVISAGE field are listed in the table 3.1 below in order to identify the differences and the added value offered by ENVISAGE project compared with the others projects.
From the analysis of those projects, we have realised that a broad part of those EU research projects focus on pragmatic approaches to resource management in a number of domains, including embedded systems, multi-core computing, grid, and cloud computing.
Many projects focus on cloud interoperability, e.g., Cloud4SOA. Several projects address linguistic abstractions for programming services on the cloud, e.g., 4CaaSt and Cloud-TM. Most of the relevant projects are concerned with one of the research aspects of ENVISAGE: virtualization, concurrent behaviour, deployment, resource analysis, contracts, services, formal verification. The novel and unique perspective of ENVISAGE is to describe virtualized resources on an abstract level and to formalize service contracts such that fully automatic analysis and verification of compliance becomes possible. Moreover, ENVISAGE goes beyond these projects by integrating resource management in the engineering of application-level services, based on formal foundations, as well as in some cases complements these projects by addressing the service design phase and by supporting automated analysis of resource management.
Here below the more relevant related research projects are listed and briefly described in order to identify the ENVISAGE added value:

[2]//www.archimate.nl/

Research project name	**Brief description**	**ENVISAGE Added Value**
CloudWave - Agile Service Engineering for the Future Internet Until October 2016	It aims at delivering innovative cloud infrastructures and tools by enabling agile development and delivery of adaptive cloud services which dynamically adjust to changes in their environment so as to optimise service quality and resource utilization. Its main innovations are about "execution analytics", "coordinated adaptation" and "feedback-driven development".	ENVISAGE aims to cover the same needs about resource usage but following a different approach based on formal methods. It is focusing specifically on the service design and analysis for virtualized services with advanced resource management support. Moreover It consider SLA at software design level.
4CaaSt - Building the PaaS Cloud of the Future Completed in August 2013.	It developed an open PaaS cloud platform which supports the optimized and elastic hosting of Internet-scale multi-tier applications. It embeds features that ease programming of rich applications and enable the creation of a business ecosystem where applications from different providers can be tailored to different users, mashed up and traded together.	ENVISAGE complements 4CaaSt by focusing on deployment in the service design phase and corresponding analysis support, e.g., resource analysis, based on a formal foundation.
CHOReOS- Large Scale Choreographies for the Future Internet Completed in September 2013	It aimed to leverage existing SOA standards and to develop dynamic and scalable choreographies QoS-aware choreographies provided by domain-experts and through the synthesis of adaptable choreographies.	ENVISAGE will integrate advanced resource management in application-level services, and focus on the engineering of virtualized services and their analysis.
Cloud4SOA - A Cloud Interoperability Framework and Platform for user-centric, semantically-enhanced service-oriented applications design, deployment, and distributed execution Completed in August 2013	It empowers a multi-cloud paradigm at PaaS level, providing an interoperable framework for PaaS developers. The system supports Cloud-based application developers with multiplatform matchmaking, management, unified application and cloud monitoring and migration. Cloud4SOA has developed a framework enabling dynamic SLA negotiation and tools that enable PaaS providers to analyse their offerings and performance and adapt the SLAs accordingly.	ENVISAGE is mainly focused on application developers focusing specifically on the service design and analysis for virtualized services with advanced resource management support. Moreover It consider SLA at software design level.

Cloud-TM - A novel programming paradigm for cloud computing Completed in May 2013	It developed a programming model for distributed applications based on atomic transactions, automating the provisioning of resources and scalability in a middleware layer.	ENVISAGE aims at leveraging deployment concerns allowing the application itself to perform high-level resource management with substantial automated analysis support.
REMICS - REuse and Migration of legacy applications to Interoperable Cloud Services Completed in August 2013.	It aimed to develop tools for a model-driven migration of legacy applications to service cloud platforms. REMICS additionally investigates existing test notations such as the UML2 test profile for their application to SOA and cloud computing.	ENVISAGE similarly includes an element of testing for deployed services, but our approach is complementary in that it is integrated in the service engineering methodology, combining a formal foundation for executable modelling with code generation.
PaaSage- Model-based Cloud Platform Upperware Untill September 2016	It aims to deliver an open and integrated platform for model based lifecycle management of cloud applications, which allows users to manage cloud resources and to have autonomic support at execution time to optimize application deployment. PaaSage will analyze and match application specifications at compile time against platform characteristics and make deployment recommendations	ENVISAGE aims at leveraging deployment in the specification language and address scalability and support agility at the application level.
CELAR-Cloud ELAsticity pRovisioning Until September 2015	It offers an innovative approach for autonomous management of hardware and software resources based on an intelligent decision-making module, which decides on the optimal expansion or contraction of allocated resources and their type in real-time. CELAR addresses elastic provisioning for cloud providers.	ENVISAGE focus on the deployment at the application level.
CloudScale - Scalability Management for Cloud Computing Until 2015	It provides service providers with tools and methods for analyzing, predicting and resolving scalability issues. A basis for this is a description language (ScaleDL) that service providers use to specify the scalability properties of basic and composed cloud services. It is developing also mechanisms to identify causes of potential SLA violations.	CloudScale addresses scalability at static time, while ENVISAGE intends to address this issue at runtime providing linguistic primitives for adaptability. Moreover ENVISAGE considers SLA issues at design level.

HARNESS - Hardware- and Network-Enhanced Software Systems for Cloud Computing Until September 2015	It deals with new virtualization techniques tailored to heterogeneous hardware and network resources for flexible allocation and reallocation and advanced techniques for optimizing cost/performance allocation trade-offs. HARNESS focuses on virtualization at the PaaS level of the Cloud stack.	ENVISAGE focus on the deployment at the application level and on SLA and its formalizations.
MODAClouds- MOdel-Driven Approach for design and execution of applications on multiple Clouds Until September 2015	It aims at delivering an advanced model-based approach to support system developers in building and deploying applications, with related data, to multi-Clouds spanning across the full Cloud stack (IaaS/PaaS/SaaS). MODAClouds addresses the cloud provider perspective and deals with issues such as migration and legacy.	ENVISAGE focus on the deployment at the application level and on SLA and its formalizations.
ARTIST - Advanced software-based seRvice provisioning and migraTIon of legacy SofTware Until September 2015	It aims at creating tools to assess, plan, design, implement and validate the automated evolution of legacy software to SaaS and the Cloud Computing delivery model.	ENVISAGE advances the ARTIST approach to testing and validation with: a) Automated analysis of resource consumption of abstract models and validation of SLAs; b) Formal verification of abstract models against requirements embodied in behavioural interfaces, service contracts, or SLAs.
Broker@Cloud - Enabling Continuous Quality Assurance and Optimization in Future Enterprise Cloud Service Brokers Until October 2015	It aims at developing a framework that will allow enterprise cloud service brokers to monitor the obligations of providers towards consumers, as well as to detect opportunities for optimizing service consumption.	ENVISAGE will provide similar monitoring systems, but using them in a complementary way because ENVISAGE techniques will enforce compliance with SLA, which will be a formal contract

Table 3.1: ENVISAGE vs others research projects

In the table below we summarize the main breakthroughs of the ENVISAGE solution with reference to the state of the art.

Main breakthrough	**Current limitations**	**ENVISAGE**
Leveraging deployment issues and resource management to the abstraction level of modeling languages	• High-level models abstract from low-level resource information • Languages with formal semantics are difficult to use and lack abstraction	• Resource management of virtualized resources by using the abstract behavioural specification language with its linguistic support • Models reflect the control and data flow of the target services • Support to automated analyses
Validation of service level agreements at design time through formal methods	• Design-by-contract only applies to functional properties; • SLA languages lack formal semantics; • Ad hoc monitoring of compliance with SLA.	• Integration of service-level agreements and object-oriented interfaces into service contracts with a firm formal semantics; • Validation of contracts based on abstract deployment scenarios and on maintaining realistic control and data flow in the models.
Novel hybrid analysis techniques for quality of service assessment	• Lack of automated methods to validate resource requirements in SLA; • Ad hoc mechanisms to trade information between static and dynamic contexts.	• A fully automatic process in which formal analysis is partly replaced with dynamic verification techniques in a systematic manner; • Combines symbolic execution and invariant reasoning with test case generation and monitoring; • The user obtains precise information on the degree of coverage and achieved precision.

Table 3.2: Main breakthrough

3.4 ENVISAGE Potential Benefits

The main benefits related to ENVISAGE project results are the following:

- accelerating the development and deployment of cloud computing and internet services" by an integrated modelling approach from requirements to deployment, with automated analysis techniques and support for code generation;
- increasing "Europe's ability to design and deliver innovative services" by integrated QoS and dynamic resource management in SLA-aware services. This enables "services with strong user engagement" because it lowers the cost of service personalization;
- supporting "better involvement of SMEs and individual researchers/ developers" through our open-source collaboratory; the ATB use case demonstrates how SMEs can profit from ENVISAGE's technology to develop innovative and individualized services;
- strengthening European software industry with the know-how to build complex services in a multilayered cloud computing continuum" by tool-based support for developing robust and efficient services with integrated QoS and dynamic resource management;
- contributing to international standardization on interoperability and SLA delivery through a foundational study of virtualization and SLA-aware services;
- leading to major economical benefits for businesses delivering virtualized services. This is because the proposed shift will reduce development costs and improve quality by detecting software defects earlier in the development process.

During this first year of project activities the ENVISAGE team has analysed the exploitable items described in section 2.2 and has identified for each one the main benefits they bring to the potential users as reported in the table below.

Exploitable item	**Potential benefits**
Simulation Tool	This tool helps developers of big data jobs get information about the potential cost and recommended cloud configuration for running the job - prior to running the job itself.
Deductive Verification Framework	This tool guarantees that a certain amount of cloud resources is sufficient to deliver a given SLA. This is important both for developers and providers of cloud services as well as, indirectly, their customers.
Test Case Generation Framework	This tools provides a set of test cases which guarantee that the selected coverage criterion is achieved. This tool reduces the amount of manually coding and maintenance of (unit) tests that a developer needs to do.
Resource Analysis Framework	This tool automatically provides developers and providers of cloud services the worst-case resource requirements for the cloud software. This can be used in software design and SLA decision making.

Collaboratory Methodology	This methodology supports developers to rapidly develop user interfaces for analysis tools. It can be used in any programming language, and programmers do not have to be familiar with complex GUI libraries.
Virtual Collaboratory	The Virtual Collaboratory is a (virtual) place where the tools developed in the context of ENVISAGE are made available to the community. External users do not need to download the collaboratory and use it locally if they do not want to and, instead, they can use it through a web interface or service. Also they do not need to learn new programming environments as the most common ones (e.g., Eclipse) will be available.
Monitoring add-ons and visualisation	This tool helps developers and providers of cloud services maintain SLAs by automatically generating monitors for rapidly evolving cloud software.
Code Generator	Developers using ABS will be able to use it also as an efficient programming language.

Table 3.3: ENVISAGE tools benefits

Chapter 4

Value Proposition

4.1 ENVISAGE Solution

As described in the previous section, ENVISAGE targets challenges inherent to virtualization, which need to be addressed for industry to effectively deliver services across distributed cloud computing resources. The two main issues identified and addressed by ENVISAGE are:

- the efficient analysis, dynamic composition and deployment of services with qualitative and quantitative service levels
- the dynamic control of resources such as storage and processing capacities according to the internal policies of the services.

ENVISAGE plans to overcome the above issues with its set of tools that enable to efficiently develop SLA-aware and scalable services, supported by highly automated analysis tools using formal methods. ENVISAGE develops a formal foundation and practical tool for the modelling, analysis, and verification of application level SLA-aware services in virtualized environments. By using ENVISAGE tools, services are able to control their own resource management and renegotiate SLA across the heterogeneous virtualized computing landscape. This will reduce management costs and optimize flexible resource usage in service delivery and will support a faster development and deployment of resource-efficient and reliable distributed cloud computing services.
The main outcome of ENVISAGE is a practical open source framework for model based development of virtualized services including:

- a behavioural specification language for describing resource aware models (extension of ABS)
- a simulator to estimate resource use in advance with visualization facilities
- a support tool for automated resource analysis, validation of SLA, code generation, and runtime monitoring of SLA for deployed services.

Through its solution ENVISAGE addresses the following needs:

Need	**Scenario**	**ENVISAGE Solution**
Modeling of resources in the design and development phase of cloud SaaS services	Virtualization makes elastic amounts of resources available to application-level services. For example, the processing capacity allocated to a service may be changed depending on the demand. Furthermore the composition of virtualized services may lead to unexpected costs and reduced quality of service. Currently, resources and resource management first appear in the deployment phase of the service engineering process, which makes it difficult and costly to fine-tune complex services to the expected runtime cost and quality of service.	ENVISAGE solution automates resource analysis allowing its users to determine at an early stage of software development the resource consumption, anticipate potential bottlenecks in the locations of the system and optimally distribute the load of work. ENVISAGE leverages ABS ability to represent environmental resources inside an executable model of the target application in an abstract manner. ENVISAGE Resource Analysis Framework takes into account the system level (e.g., the fact that distributed nodes communicate and transmit data among them, post tasks to other nodes queues, etc.) and the deployment descriptions contained in the models (such as the use of scheduling policies). Thus, the framework obtains more precise results for each component and also information involving several components like data transmission and spawned tasks.
Management of Service Level guarantees at design and at runtime	Currently, a legal contract between the service owner and the cloud provider defines the service-level agreement regulating resource provisioning to the service. However, the legal contracts are not integrated in the programming models, which severely restricts the developer in fine-tuning the service to the available virtualized resources and prevents the system to adapt itself to possible failures.	ENVISAGE solution provides a set of tools to monitor at runtime the SLAs. In ENVISAGE service-level guarantees are first class concepts and part of the design model of the application and also part of the deployment model. A monitoring system ensuring adaptability to failures and to renegotiations of service-level agreements.

Simulation of big data jobs prior to costly runs	Often to run applications on the cloud requires lots of expensive resources (e.g. CPU and storage) an example is the running of big data jobs. It would be useful to experiment, tune parameters before running the actual applications. This can lead to a more rapid and affordable way of getting to a good or optimal configuration before the actual run. Support for this kind of experimentation will also offers the opportunity to perform more risky experiments through simulation with outlier configurations that you would not have done at a larger scale.	ENVISAGE solution offers a modelling and simulation tool to estimate resource usage in advance (time and cost), and also provide recommendation of which cloud configuration (e.g. Hadoop cluster machine types) to use for cost-efficient results. In Envisage application-level services will be simulated in the context of an abstract deployment scenario which allows the fine-tuned control of virtualized resources. The innovation is this simulation combined with utilization of all software quality related features in ABS language and supporting tools (e.g. unit test creation, static analysis for resource analysis and concurrency) to ensure improved software quality of big data jobs prior to their run through simulation (e.g. to reduce risk of jobs failing during costly runs).
Choose the most cost efficient configuration	When developing applications to run on a distributed environment several configuration options may be possible and it is useful to know which is the optimal one to get the most from a distributed environment and to reduce as much as possible. On the market there are resource analyzers (like COSTA [4], SPEED [28], RAML [23]) which mainly deal with traditional applications and even if there are recent extension [3] to considers distributed systems they are not very precise. This is mainly because these tools ignore the underlying deployment configurations and service interactions.	ENVISAGE Simulation Tool is a simulation environment which allows the rapid prototyping of models in different deployment scenarios and with different load balancing strategies. ENVISAGE provides also a cost analysis framework which is able to automatically understand the deployment configuration and the interactions among services, and to automatically infer the cost by attributing the corresponding cost to each service. In this way ENVISAGE provides recommendations on which is the best cloud configuration in order to give its users the opportunity to choose the optimal and most cost efficient configuration.

Higher quality of the software produced	Software quality is one of the pivotal aspects of a software development company. Furthermore the longer software defects go undetected the more expensive they are to fix. Typical bugs that occur in application to run on virtualized environment are concurrency-related (e.g. for interactive code in apps or servers), or resource-overuse related (e.g. for big data jobs such as Mapreduce/Hadoop).	ENVISAGE supports higher quality of the code and applications produced Through *formal verification* and the generation of a set of test cases which guarantee that the selected coverage criterion is achieved and by shifting the validation phase of virtualised services to the design time.

Table 4.1: Needs covered by ENVISAGE solution

By mastering virtualization in the engineering of services, ENVISAGE has the potential to address the above needs and significantly improve the competitiveness of SMEs, and profoundly influence business ICT strategies in all sectors.

4.1.1 Potential Customers

The results of the ENVISAGE projects are mainly addressed to developers of applications for virtualized environments. Nevertheless, we have identified two main customer categories, that is the application developers and the cloud providers who could provide ENVISAGE solution integrated as a whole or in part into their development tools.
In the table below, the potential customers which are included in these categories and the potential benefits they have by adopting ENVISAGE solutions are described:

Customer Category	Customers	Benefits
Application developers and system integrators	SaaS providers, Service Application Developers, SME, Big company involved in application development for cloud environment, Big Data application developers,ect.	• automation of resource analysis allowing its users to determine at an early stage of software development the resource consumption, anticipate potential bottlenecks, optimally distribute the load of work. • monitor at runtime the SLAs ensuring adaptability to failures and to renegotiations of service-level agreements. • estimation of resource usage in advance (time and cost). • recommendation of which cloud configuration to use to choose the optimal and most cost • improved software quality of big data jobs prior to their run through simulation. • rapid prototyping of models in different deployment scenarios and with different load balancing strategies • higher quality of the code and applications produced
Cloud providers	In particular PaaS providers	• Better profiles of the resource needs of the client layer help cloud providers to avoid over-provisioning to meet their SLAs. • Better usage of the resources means more clients can be served with the same amount of hardware in the data centre, without violating SLAs and incurring penalties.

Table 4.2: Potential Customers and Benefits

Chapter 5

Exploitation Plan

5.1 Initial Joint Exploitation Plan

At this early project time, two main ways to exploit project results are been identified. The first is related to the ABS exploitation while the second is linked to the ENVISAGE Collaboratory tool development. At the current stage of the exploitation activities the project partners have decided to follow an open source license approach to exploit its results. In the next months the Consortium will discuss on the more appropriate licenses to be adopted for the technical project results both as components and as whole solution.

5.1.1 Joint Exploitation leveraging on ABS eco system

Currently ABS and the ABS tools are open source. The ABS tools are stable and constantly maintained. There is a mailing list for support requests. The ABS language and tool set has been used to model and analyze complex industrial software [5]. The ABS language and tool set is conceived as a long term project with the explicit goal to build up an international community of users and contributors. The ENVISAGE partners UIO, CWI and TUD are committed to develop and maintain ABS even beyond the lifetime of ENVISAGE. A number of planned projects will also use ABS, for example, TUD plans to use ABS to model railway signalling logic.
Several research groups that were not involved with ABS have started to pick it up: for example, at INRIA Sophia Antipolis [22] or University of Indonesia [30]. Some of these researchers interested in ENVISAGE technology have been invited to the project's Scientific Advisory Board. At University of Indonesia currently a number of ABS extensions are developed that will greatly increase its usability in industrial context: these include a relational database interface via ORM, ABSUnit testing, an http server, a web application framework. Several companies that are not associated with ENVISAGE have expressed their interest in ABS technology, including German Railways (Deutsche Bahn Netz AG) and Bosch GmbH.

5.1.2 Joint Exploitation leveraging on ENVISAGE Collaboratory

ENVISAGE Collaboratory is a (virtual) place where the tools developed in the context of ENVISAGE are made available to the community. The Collaboratory is going to support the building and management of the user community for ENVISAGE methodology and tools. Users will be able to experiment ENVISAGE tools but also use them in different forms, e.g., as services, through a web-interface, by downloading to use locally, etc. The collaboratory will also allow users to share their experience and provide feedback. It will offer an easy way for programmers to integrate their own tools (through the easy-interface language showed in the Figure 5.1 to the top right) giving the community members the opportunity to actively contribute to the ENVISAGE community.

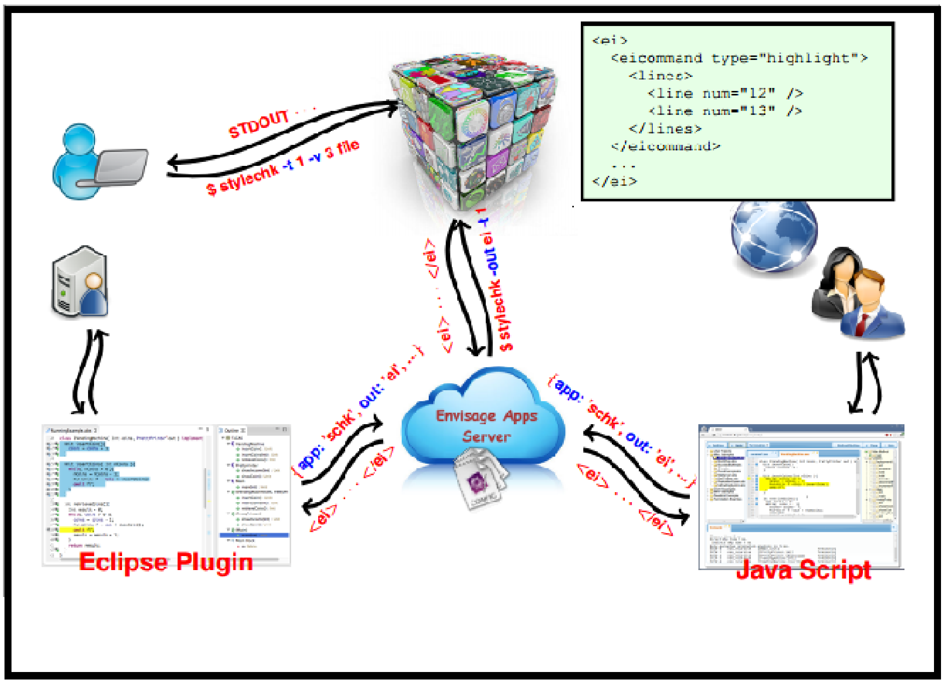


Figure 5.1: ENVISAGE virtual Collaboratory–an APPs server and a GUI Language

Through the Collaboratory we will deliver open-source software components for developers which enable community-based software development, as well as user-friendly interfaces for users. Community-based software development is an initiative that gives community groups control of the software development process, resources, and decision making. The underlying assumption of community-based software projects is that communities are the best judges of how to best improve the software, and, if provided with adequate tool support and information, they can organize themselves to provide for their immediate needs.
The ENVISAGE Consortium aims at promoting community-based software development for its technological results by providing:

- *Information and tools to users*: our tools will be usable through a web-based interface. Plug-ins to a development framework such as Eclipse will provide an intuitive user interface to the tools, examples, and documentation.

- *Information and tools to developers*: ENVISAGE technology is made available as open-source software through the Collaboratory, including documentation so that developers can easily extend and adapt the technology to their needs.

- *Information and tools to researchers*: We will maintain a list of open research challenges related to SLA-aware virtualized services and their analysis.

Pillar in our exploitation strategy is the transition of the ENVISAGE collaboratory into an open-source community-based software development project. Consortium Partners (TUD, UCM, and UIO) are committed to keep the Collaboratory online and running also after the project.
The success of the joint exploitation plan is supported by a well-balanced consortium which is composed both by academy and industry, large enterprises and SMEs, four EU member states and one non-EU member state. More in detail it includes 5 academic (UIO, CWI, TUD, BOL and UCM) and 3 industrial partners (one SME) located in Germany, Italy, the Netherlands, Norway, and Spain. The industrial partners are coming from three complementary areas where an efficient use of the benefits coming from virtualization techniques i.e. reliability and an efficient control of resources will become a defining parameter of success:

search technology (ATB), e-commerce (FRH) and cloud provisioning and migration of services (ENG).
In the table below we summarize the domains of interest of each partner which reflect the areas covered by the Consortium during its dissemination and exploitation activities. More information on each partners' exploitation plan can be found in the following section.

Partner	**Domain**
UIO	Modeling concurrent systems, object orientation, verification, formalization of contracts
CWI	Semantics and verification of concurrent systems
TUD	Resource analysis, Static analysis, formal modeling, test case generation
BOL	Components, behavioral types, concurrency, service-oriented systems, contracts
UCM	Resource analysis, test case generation, partial evaluation
ATB	Search technology, cloud computing, SaaS, technology commercialization
FRH	E-commerce software, real-time personalization, monitoring, SaaS, PaaS
ENG	Cloud computing, service-level agreements, SaaS, IaaS

Table 5.1: ENVISAGE Partners Domains

5.2 Individual Exploitation

As anticipated in the previous section, ENVISAGE exploitation strategy builds upon the specific complementary expertise of all partners. In this section each partner provides information regarding the technical outcomes, skill and competences acquired from the project activities, as well as its individual exploitation plan.

5.2.1 Universitetet I Oslo (UIO)

Technical Outcomes (sw, components, integrated system, hardware etc)	
ABS scalable interpreters	Technology for the simulation and rapid prototyping of large models, running on multiple physical or virtual machines.
Competence and Skills	
Ability to develop abstract behavioural specification languages	Approach to the design of executable formal modelling languages combining object-orientation, asynchronous communication, concurrency, and resource management.
Ability to decompose non-functional properties as service contracts	Design-by-contract approach to the specification of non-functional properties in terms of behavioural interfaces
Developing expertise in analysis techniques for abstract behavioural specifications	Includes deductive verification techniques, type-based analyses, symbolic execution, testing, and visualization of designs.

Hybrid analysis techniques	Ability to combine diverse and complementary analysis technique.

Individual Exploitation Intentions
We plan to use the technical outcomes and competences developed in Envisage for future research projects and initiatives, both for applied research in collaboration with industrial partners and in further development of technologies and competences through nationally and European funding programs. In particular, we will use and refine these technologies and competences in the recently granted EU H 2020 project HyVar, the EU FP7 project UpScale, the Norwegian project Cumulus. A recently funded Norwegian Center for Research-driven Innovation (www.sirius-labs.no) has Envisage as one of its technological instruments and E. B. Johnsen as one of its main drivers. The Center focuses on innovation for scalable data access and includes major international companies within the Oil & Gas sector. There has also been initiatives for research funding to apply Envisage technologies for the gradual virtualization of business software together with the main players in this sector on the Norwegian market (Visma, SuperOffice, etc). More in the long term, we are in dialogues to establish consortia looking at energy-aware computing using Envisage techniques and technologies, targeting H 2020.

5.2.2 Stichting Centrum Voor Wiskunde en Informatica (CWI)

Technical Outcomes (sw, components, integrated system, hardware etc)	
Efficient backends for the ABS language.	Implementation of the ABS language in both Java and Haskell to support the generation of production code.
Competence and Skills	
Formal semantics.	A formal semantics will be used to prove formally the correctness of the Java and Haskell implementations.
Formal verification.	Both backends will be integrated into a model-based development environment which supports various analysis techniques.
Real-time.	Both backends will feature real-time mechanisms.

Individual Exploitation Intentions
We plan to create a developers community for the ABS programming language. We further will use the ABS backends for simulation of biosystems.

5.2.3 Technische Universitaet Darmastadt (TUD)

Technical Outcomes (sw, components, integrated system, hardware etc)	
KeY ABS	Deductive verification system for distributed ABS models that allows us to verify functional as well as specific resource related properties.
CoFloCo	A static analysis tool that infers symbolic complexity bounds on the resource consumption of ABS models.
Type analysis of delta-oriented Software Product Lines (SPL)	A compiler component to check type safety of delta-oriented Software Product Lines implemented in ABS. This family-based analysis approach will ensure that the products of the SPL are well-typed by analysing a combined representation of all possible products in one run. We expect a major improvement in efficiency, making ABS more suitable for the development of industrial-size SPLs.
Competence and Skills	
Ability to formalise (parts of) SLAs in logic	Formalising SLAs in logic is a precursor for analysing ABS models/programs whether they can provide the obligations of an SLA.
Ability to deductively analyse distributive systems	Verification of distributed systems allows us to mathematically prove the absence of program errors and gives the strongest possible guarantees that a program satisfies its contract (functional properties or resource consumption).
Developing expertise in the automatic computation of symbolic upper bounds for resource consumption of distributed systems	Analysis of ABS models/programs by computing upper bounds that limit the resource consumption such as memory/CPU consumption.
Knowledge about a range of analysis techniques and how to combine them to obtain improved analysis results or to achieve a higher degree of automation.	To be able to predict the behaviour of a system with respect to resources etc. using only one analysis technique is not sufficient. Instead the precision and/or automation of the analyses can be increased by combining different techniques. In particular we enhances our knowledge about complementary analyse methods.

Individual Exploitation Intentions
We plan to use the technical outcomes and skills in future research projects with industrial partners to model and analyze distributed systems. In the planning stages is e.g. a project to model and analyze the signalling logic of the German railways network. The obtained skills are intended to be applied to different kinds of (distributed) systems e.g. embedded systems and to ensure that programs can operate with limited resources. Also of interest is the transfer to other resource kinds than memory or runtime. Of particular interest is the analysis of energy usage as a necessity for sustainable development.

5.2.4 Alma Mater Studiorum-Universita Di Bologna (BOL)

Technical Outcomes (sw, components, integrated system, hardware etc)	
Programming Languages for Cloud Computing	In order to program virtualized services, mainstream programming languages must be extended with primitives that deal with elastic managements of resources. It is necessary that these primitives have a semantics for supporting formal analyses.
The formalization of behavioural interfaces with quality of services informations.	In order to reason about virtualized services in a compositional way one needs informative types recording the abstract behaviours with respect to virtualized resources. These types must have a formal semantics in order to support mathematical proofs.
Algorithms for verifying properties of behavioural interfaces automatically and efficiently.	Once behavioural interfaces have been defined, it is not possible to ask to programmers to learn an additional language, as well as it is not possible to ask for verifying properties using such interfaces. Therefore tools must be provided that (1) extracts automatically behavioural interfaces out of (ABS) codes and (2) verify the correctness of such interfaces with respect to sensible properties defined by the SLA.
Competence and Skills	
Concurrent Object-oriented Programming Languages	Syntax, semantics, virtual machines.
Static analysis	Type systems with expressive types (called behavioural) and abstract interpretation of programming languages.
Prototype implementations	Implementations of languages, of verifiers, of inference systems.

Individual Exploitation Intentions
BOL will use the formal techniques developed for verifying properties of virtualized systems in current and future EU and national projects, other than Envisage, in order to investigate their uses in other contexts, such as energy efficient computing.

5.2.5 Universidad Complutense De Madrid (UCM)

Technical Outcomes (sw, components, integrated system, hardware etc)	
Testing Framework	New strategies and heuristics for pruning redundant state-exploration when testing actor systems by reducing the amount of unnecessary non-determinism. Support for guiding the exploration towards specific task interleavings and, specifically, those that produce deadlocks.
Symbolic Execution	New termination criteria for concurrent programs which ensure finiteness of symbolic execution and at the same time provide coverage criteria for the executed code.
aPET Testing Tool	A testing tool for concurrent objects that can be used both for dynamic and static testing by means of symbolic execution. Additionally it can be used for automatic and high-quality test-case generation.
Resource analysis	A novel static resource analysis to infer bounds in ABS models with interleaved computations in nodes which relies on May-Happen-in-Parallel information.
Priority-based MHP	An extension of the MHP analysis to use priorities of the different tasks invocations and improve the obtained parallelism information.
Peak cost analyzer	Inference of precise bounds on the peak of the resource consumption (executed instructions, memory) of ABS models.
Inference of transmission data sizes	Static analysis to infer the amount of data that a distributed system may need to transmit among the nodes in the system.
Performance indicators	New indicators of the performance of the distributed system which allow automating the comparison of different settings of the system and guiding the developer towards finding the optimal configuration.
Easy Interface	This software is the core of the Envisage Virtual Collaboratory. It consists of an APPs server and several development environment clients (e.g., web-interface, eclipse-plugin). This software helps programmers to rapidly build graphical user interfaces (GUI) for their applications with minimal effort, in particular it does not require the user to be familiar with GUI libraries or web programming, and can also be used for programs written in any programming language since it is text-based.
Competence and Skills	

Ability to debug concurrent systems	The techniques will be applied to speed up tools dedicated to dynamic and static testing. As a consequence the use of such tools will help develop reliable software and will increase the productivity in the industry. Besides, the testing techniques will be used to ensure the partial correctness of the implementations developed in ENVISAGE.
Knowledge about the scheduling policies applied to ABS models and its impact in parallelism information	The scheduling policies used in the different tasks affect greatly the information that the MHP analyses can infer. Using this system level information is is possible to obtain more precise results.
Ability to integrate MHP information into other static analysis	A better understanding of how the MHP information can be integrated to improve other static analyses like termination or resource usage. This MHP information of ABS models can be used to discard unfeasible interleavings, therefore improving the overall results.
Knowledge of the amount of data transmitted among the locations a distributed system.	A worst-case cost estimation of the data transmissions in a distributed system using cost analysis
Formal techniques to guide the configuration process of distributed systems	Knowledge on performance indicators, such as communication among nodes, level of distribution and load balance in the system locations
Development of generic GUI	During the development of the Easy Interface software, members of UCM acquired important knowledge and skills in developing generic GUIs text-based languages, and in developing web-based environments. These skills are important for extending Easy Interface to support more sophisticated interfaces in the future.

Individual Exploitation Intentions
UCM will intensively use the formal techniques developed for the testing and analysis of distributed systems in current and future EU and national projects, other than Envisage, in order to investigate their uses in other contexts (such as for the analysis of HPC applications). The ability to infer the peak cost of the nodes in a distributed system is directly applicable to industry, for instance in the context of virtualization as used in cloud computing, as the peak cost allows estimating how much processing/storage capacity one needs to buy in the host machine, and thus can help reduce costs. UCM will intensively use the Collaboratory software in current and future EU and national projects, other than Envisage, in order to build platforms similar to the Envisage Virtual Collaboratory. Easy Interface will be released as a free open-source software, independently from the Envisage Virtual Collaboratory, and will be maintained by UCM. This is expected to help other projects and individual users in promoting their research tools by providing easy access to them.

5.2.6 Atbrox AS (ATB)

Technical Outcomes (sw, components, integrated system, hardware etc)	
Simulation tool (D1.4.1 and D1.4.2)	Atbrox provides log data and expertise from developing and running actual big data jobs (Mapreduce and later Spark - using the Elastic Mapreduce Hadoop-based Cloud Service from Amazon Web Services - that can be used as input to tune and shape the modelling and simulation tools developed by UiO. These tools can be part of the collaboratory and be used by others to get estimates of costs and configuration for cloud big data jobs prior to actually running them and taking on the cost (i.e. decision support wrt big data jobs)
Code generation from ABS for at least one widely used or fast growing programming languages, e.g. Python or Go for backend development, or Swift for mobile app development (iOS).	This can ensure that critical parts of code that is used in production settings has higher quality since it can be analysed using ABS tools, in particular wrt resource analysis and concurrency. Python is one of the fastest growing programming languages (rapid development and a large set of libraries and tools), e.g. used by large mobile and Internet services such as Dropbox, Instagram, Path, Quora, Youtube and Pinterest. Go - developed by Google - is a fast growing language for developing scalable backends with similar characteristics as Erlang. Swift - developed by Apple - is replacing Objective-C as the language to develop mobile apps for iOS devices.

Commercial packaging/web services for the ENVISAGE Collaborative	Software developments lags behind most other technical services in terms of quality requirements, this is likely to change since the impact of errors in software is increasing (e.g. with Internet of Things etc.). Providing services for increasing software quality, e.g. competing with companies such as Coverity - urlhttp://www.coverity.com/why-coverity/ - urlhttp://www.coverity.com/services/ could be a goal for the ENVISAGE commercial dissemination.
Competence and Skills	
Learn to use formal methods and the ABS programming language to model cloud service and mobile app code	Atbrox team has little background in formal methods for use in practical software engineering, and ENVISAGE project outcomes can help us gain from using formal methods.
Developing expertise in using state-of-the-art static analysis methods for ABS to improve software quality (resource)	Static analysis tools for programming languages Atbrox typically use today (e.g. Python, Objective-C, Swift and C++) are very limited compared to what ABS supports, in particular wrt resource usage and concurrency. The most frequently occurring and serious bugs we typically experience in day-to-day development are typically concurrency-related (e.g. for interactive code in apps or servers), or resource-overuse related (e.g. for big data jobs such as Mapreduce/Hadoop). Our experience is that concurrency bugs are very common throughout the software developing industry.
Developing expertise in using tools for automated unit test creation (D3.5 from UCM)	Unit tests are necessary, but they are costly to develop, and probably even more costly to maintain over time. By using tools to generate them rather than writing them ourselves, we can increase productivity but at the same time ensure that quality is high since software we create get tests.

Individual Exploitation Intentions
Atbrox creates both server-side (in the cloud) and mobile applications with heavy use of concurrency in both cases. Bugs we experiencing are typically related to resource overuse (e.g. out-of-memory) and concurrency (in particular within mobile apps). Using ABS for modelling critical concurrency-related code could make us more productive and provide higher quality to our customers. We believe this competency can be used to perhaps charge higher rates, or provide new consulting related products related to going into existing software engineering teams and training them.

5.2.7 Fredhopper B.V. (FRH)

Technical Outcomes (sw, components, integrated system, hardware etc)	
Distributed Monitoring System	A modular monitoring system composed of metrics data backend, HTTP interaction layer, and visualization component.
A generic formal model of QoS metrics	• The model is independent of any technical SLA spec • The model uses ABS model as a basis for distributed and concurrent communication • The model can be generated from other SLA models to capture the QoS metrics .
Monitoring SDK API	• Allows modeling of any SLA metrics • Developed on top of ABS with its Java API • Allows custom definition of a metric on a layered architecture to integrate with the monitoring system.
Competence and Skills	
Modelling a complex and expert monitoring system	Transferring a current operation model on monitoring system from a technology POV to a formal design approach
Deploying the research and deployment in a large environment operated by SDL Fredhopper	Utilizing the results of the research and the system that is developed in this project and bringing them back to the operation environment as a feedback loop

Individual Exploitation Intentions

5.2.8 Engineering - Ingegneria Informatica S.p.A. (ENG)

Technical Outcomes (sw, components, integrated system, hardware etc)	
Resource Pool Manager	It is a component for the elastic management of the computational resources utilised by ENG's ETICS service. ETICS is a web-based service for the execution and quality assurance of builds and tests for distributed, multi-language, multi-platform software. In the current incarnation of ETICS, the underlying pool of computational resources is managed manually. Consequently through the adoption of Envisage methodology a 'Resource Pool Manager' will be developed, as a proof of concept prototype, for make automatic the elastic management of the ETICS computational resource pool.
Competence and Skills	
How to optimize the allocation of computational resources	Thanks to the experience with the Envisage Resource Analysis Framework ENG will learn how to anticipate potential bottlenecks in the locations of the system and to optimally distribute the load of work in order to fulfil the service contracts of the components services.
How to Improve SLA management	Thanks to the experience with the KeY-ABS deductive verification system for ABS ENG will learn how to better specify SLAs in terms of the restriction between resource consumption, resource provision and load balancing.
Programming skill with the ABS language	Thanks to the development of one of the project's case study ENG will acquire general skill with the ABS language.
How to formally specify QoS requirements.	Thanks to the contribution to T2.2 ENG will acquire the skill to formally specify QOS requirements and to formally prove (whenever possible) the compliance of an ABS system design with those requirements.

Individual Exploitation Intentions
The Engineering Group is present in the IT outsourcing market with a network of data centres (Pont Saint Martin, Turin, Padua) adopting advanced technologies to ensure high service quality and security. The services offered through the data centres cover all the needs of Business processes, ranging from Application and Information Systems management; Business process outsourcing, including remote desktop management, server consolidation, virtualization and disaster recovery; consultancy services. Nowadays, the IT operation division is servicing more than 200 customers with 5000 servers, 40000 remote desktops and 1000 remote connections. The data centre of Pont Saint Martin is heavily investing in green technologies, including geothermal solutions for cooling the resources. The outcomes, the skills and competences acquired within ENVISAGE project will give ENG the opportunity and ability to change the way to develop application/services for cloud environments as well as to operate on the cloud-service provisioning. In particular the project results will enable ENG to: • Optimize the allocation of computational resources to satisfy customers' demand, ENG will be able to increase the revenue by serving more customers without necessarily increasing the computational resources. • Increase automation of the application deployment process will significantly reduce the time to operate services. This not only reduces personnel costs for operating the service, it enables ENG to change the current business model to also offer fine-grained and short-time capabilities (cloud resources). Changing the pricing model in this way will reduce the costs for customers to make cloud services affordable to larger markets, potentially increase the overall revenue. • Improve SLA management through formalization, analysis and test techniques, will enable ENG to better satisfy customers' demand and to mitigate the risk of SLA violation. This directly impacts on reducing penalty costs. Indirectly it also impacts the quality of service perceived by ENG customers, empowering ENG to attract more customers.

Chapter 6

Conclusions

In this deliverable we have provided a description of the exploitable items which were identified in the first year of project activities. For each item we have analysed and presented the innovative features with reference to the state of the art. We have also provided a survey on the target cloud market segments identified as more suitable for EVISAGE solution. These market segments comprise the SaaS, PaaS and IaaS and the results show breeding ground for ENVISAGE thanks to the rising trends for cloud computing adoption and investments. In the deliverable we have presented the initial gap analysis looking at potential competitors and identifying a set of potential benefits offered by ENVISAGE. Building on the results of these first activities we have identified what ENVISAGE solution is and what issues it aims to overcome, as for example the efficient analysis, dynamic composition and deployment of services with qualitative and quantitative service levels and the dynamic control of resources such as storage and processing capacities. In brief ENVISAGE provides its customers with a set of tools that enable to efficiently develop SLA-aware and scalable services, supported by highly automated analysis tools. The main customer groups identified for ENVISAGE solution are application developers, system integrators and cloud providers (PaaS providers) for which we have identified a set of potential benefits, as presented in section 4.3. The final chapter (chapter 5) is dedicated to the initial exploitation plans and intentions both from the point of view of the whole Consortium and of the single partners. In the following months, the WP5 team will continue working on the market analysis and exploitation activities in order to refine the results presented in this deliverable. An update of the results of this second phase of activities will be given in the following version of the deliverable (D5.7 - Business Strategy & Revised Exploitation Plan).

Bibliography

[1] *The ABS Language Specification*, 1.2.0 edition, April 2013. `http://tools.hats-project.eu/download/absrefmanual.pdf`.

[2] Accenture. "2014 north america digital mining survey,". Research report, 03 2014. Available from`http://www.accenture.com/us-en/Pages/insight-north-america-digital-mining-survey.aspx`.

[3] Elvira Albert, Puri Arenas, Samir Genaim, Miguel Gómez-Zamalloa, and Germán Puebla. Cost Analysis of Concurrent OO programs. In Hongseok Yang, editor, *Proceedings 9th Asian Symposium on Programming Languages and Systems (APLAS 2011)*, volume 7078 of *Lecture Notes in Computer Science*, pages 238–254. Springer-Verlag, 2011.

[4] Elvira Albert, Puri Arenas, Samir Genaim, Germán Puebla, and Damiano Zanardini. Cost Analysis of Object-Oriented Bytecode Programs. *Theoretical Computer Science*, 413(1):142–159, 2012.

[5] Elvira Albert, Frank S. de Boer, Reiner Hähnle, Einar Broch Johnsen, Rudolf Schlatte, Silvia Lizeth Tapia Tarifa, and Peter Y. H. Wong. Formal modeling of resource management for cloud architectures: An industrial case study using Real-Time ABS. *Journal of Service-Oriented Computing and Applications*, 8(4):323–339, 2014.

[6] Elvira Albert, Reiner Hähnle Frank de Boer, Einar Broch Johnsen, and Cosimo Laneve. Engineering virtualized services. In M. Ali Babar and Marlon Dumas, editors, *2nd Nordic Symposium on Cloud Computing & Internet Technologies (NordiCloud'13)*, pages 59–63. ACM Press, 2013.

[7] Victor R. Basili Barry Boehm. *"Software Defect Reduction Top 10 List,"*, volume Vol. 34, No. 1. IEEE Computer Society, Jan. 2001.

[8] Marco Bernardo, Ferruccio Damiani, Reiner Hähnle, Einar Broch Johnsen, and Ina Schaefer, editors. *Executable Software Models: 14th International School on Formal Methods for the Design of Computer, Communication, and Software Systems, Bertinoro, Italy*, volume 8483 of *Lecture Notes in Computer Science*. Springer-Verlag, June 2014.

[9] R. Buyya C. S. Yeo S. Venugopal J. Broberg and I. Brandic. *Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. Future Generation Computer Systems.* 2009.

[10] Richard Bubel, Antonio Flores Montoya, and Reiner Hähnle. Analysis of executable software models. In Bernardo et al. [8], pages 1–27.

[11] Newcombe C. Why amazon chose tla+. volume 8477 of *in Abstract State Machines, Alloy, B, TLA, VDM, and Z Lecture Notes in Computer Science*, pages 25–39, 2014.

[12] Cisco. "midsize enterprises leading the way with cloud adoption". Research report, 2014. Available from `http://share.cisco.com/cloudadoption/`.

[13] Emerson. Data center 2025: Exploring the possibilities. Technical report, 2014. Available from http://www.emersonnetworkpower.com/en-US/Latest-Thinking/Data-Center-2025/Documents/002401_DataCenter2025Report_HR_INTERACTIVE.PDF.

[14] IDG Enterprise. "worldwide saas and cloud software 2014–2018 forecast and 2013 vendor shares". Technical report, 07 2014. Available from http://www.idc.com/getdoc.jsp?containerId=249834.

[15] IDG Enterprise. "worldwide competitive public platform as a service 2013–2017 forecast". Technical report, 11 2013. Available from http://www.idc.com/getdoc.jsp?containerId=243315.

[16] IDG Enterprise. "computerworld 2014 state of the enterprise". Technical report, 2014. Available from http://www.idgenterprise.com/report/computerworld-2014-state-of-the-enterprise.

[17] Eurostat. Ict usage in enterprises in 2014, cloud computing services used by one out of every five enterprises in the eu28. Technical report, 9 December 2014. Available from http://ec.europa.eu/eurostat/documents/2995521/6208098/4-09122014-AP-EN.pdf/627ddf4f-730a-46ca-856b-32532d8325c5.

[18] Gartner. "market trends: Platform as a service, worldwide 2013-2018". Technical report, 2013. Available from http://www.gartner.com/document/.

[19] Reiner Hähnle. The Abstract Behavioral Specification language: A tutorial introduction. In Marcello Bonsangue, Frank de Boer, Elena Giachino, and Reiner Hähnle, editors, *International School on Formal Models for Components and Objects: Post Proceedings*, volume 7866 of *Lecture Notes in Computer Science*, pages 1–37. Springer-Verlag, 2013.

[20] Reiner Hähnle, Michiel Helvensteijn, Einar Broch Johnsen, Michael Lienhardt, Davide Sangiorgi, Ina Schaefer, and Peter Y. H. Wong. HATS abstract behavioral specification: the architectural view. In Bernhard Beckert, Ferruccio Damiani, Frank de Boer, and Marcello M. Bonsangue, editors, *Proc. 10th International Symposium on Formal Methods for Components and Objects (FMCO 2011), Torino, Italy*, volume 7542 of *Lecture Notes in Computer Science*, pages 109–132. Springer-Verlag, 2013.

[21] Reiner Hähnle and Ina Schaefer. Adaptable and evolving software for eternal systems — (track summary). In Tiziana Margaria and Bernhard Steffen, editors, *Leveraging Applications of Formal Methods, Verification and Validation. Technologies for Mastering Change — 5th International Symposium, ISoLA 2012, Heraklion, Crete, Greece*, volume 7609 of *Lecture Notes in Computer Science*, pages 1–3. Springer, October 2012.

[22] Ludovic Henrio and Justine Rochas. Declarative scheduling for active objects. In *Proc. of the 29th Annual ACM Symposium on Applied Computing*, SAC '14, pages 1339–1344. ACM, 2014.

[23] J. Hoffmann and M. Hofmann. Amortized resource analysis with polynomial potential. In *The 19th European Symposium on Programming (ESOP'10)*, volume 6012, pages 287–306. Springer, 2010.

[24] IHS. Cloud- related spending by businesses to triple from 2011 to 2017. Technical report, 2014.

[25] KPMG International. "breaking through the cloud adoption barriers". Research report, 2012. Available fromhttps://www.kpmg.com/SG/en/IssuesAndInsights/ArticlesPublications/Documents/Advisory-ICE-Breaking-through-the-Cloud-Adoption-Barriers-Glob.pdf.

[26] Einar Broch Johnsen, Reiner Hähnle, Jan Schäfer, Rudolf Schlatte, and Martin Steffen. ABS: A core language for abstract behavioral specification. In Bernhard Aichernig, Frank S. de Boer, and Marcello M. Bonsangue, editors, *Proc. 9th International Symposium on Formal Methods for Components and Objects (FMCO 2010)*, volume 6957 of *Lecture Notes in Computer Science*, pages 142–164. Springer-Verlag, 2011.

[27] Dimosthenis Kyriazis. Cloud computing service level agreements, exploitation of research results. June 2013.

[28] S Gulwani K K Mehra and T M Chilimbi. Speed: Precise and efficient static estimation of program computational complexity. volume In The 36th Symposium on Principles of Programming Languages (POPL 09), pages 127–139. ACM, 2009.

[29] European Commission Expert Group Report. The future of cloud computing: Opportunities for european cloud computing beyond 2010. 2010.

[30] Maya Retno Ayu Setyautami. Applying uml profile and refactoring rules to create software product line from uml class diagram. Master's thesis, Faculty of Computer Science, Universitas Indonesia, 2013.

[31] Peter Y. H. Wong, Elvira Albert, Radu Muschevici, José Proença, Jan Schäfer, and Rudolf Schlatte. The ABS tool suite: modelling, executing and analysing distributed adaptable object-oriented systems. *Journal on Software Tools for Technology Transfer*, 14(5):567–588, 2012.