NATRES' White-paper

Version 1.0

Authors:

Dana Petcu, West University of Timisoara, Romania
Lara López, F. Javier Nieto, Atos Spain SA, Spain
Iakovos Mavroidis, Foundation for Research and Technology - Hellas, Greece
Roberto Bruschi, CNIT, Italy
Zhiming Zhao, University of Amsterdam, the Netherlands
Constantinos Vassilakis, Panagiotis Gouvas, UBITECH, Greece
Isabel Matranga, Engineering Ingegneria Informatica SpA, Italy
Daniele Pavia, Engineering Ingegneria Informatica SpA, Italy
Andy Edmonds, Thomas M. Bohnert, Zürich University of Applied Sciences, Switzerland
Maria Fazio, University of Messina, Italy
Philipp Leitner, University of Zurich
Eliot Salant, IBM Research - Haifa

March 6th, 2016

CONTENT

1. Scope

1.1 Aim

This open-access document aims to support the activities of the cluster of projects and actions on Cloud partially funded by the European Commission through the FP7-Collaboration-ICT and H2020-LEIT-ICT programmes. It identifies the current research approaches and newest technologies in the field of resource and service management in Cloud environments, as well as the gaps that are not yet fulfilled by them. This information is considered essential to identify future research and development directions, as well as to help the initiation of new actions and projects in the field of resource and service management.

1.2 Scope

The document is reflecting the opinions of the representatives of the projects involved in the cluster of projects, respectively ARCADIA, BEACON, CloudLightning, CloudWave, ClouT, DICE, ENTICE, INPUT, Mobile Cloud Networking, MODAClouds, RAPID, SPECS, SWITCH.

The cluster "Novel approaches and technologies for resource and service management" (NATRES) intends to be a forum for discussing the current research and innovation challenges encountered at infrastructure-as-a-service level generated by the desire to improve the user experiences and the efficient use of the available resources. The current trends are including the integration of special devices from high performance computing ones to mobile devices, the design of decentralised service-oriented systems, the improvement of the virtualization technologies, the overcome of portability and interoperability issues, or the automation the organisation and management of the back-end resources. Cloud-based applications from the fields of Internet-of-Things and Big Data are expected to challenge the new services.

Details about the cluster activities, aims and results are available at:
https://eucloudclusters.wordpress.com/new-approaches-for-infrastructure-services/

1.3 Organisation

The document is organized as follows. The next section presents the approaches and technologies developed by the cluster actions in the field of resource and service management as well as their expected impact on the market. The third section is exposing the gaps identified by the actions in the theoretical and technological solutions that can be subject of future actions. Conclusions are provided in the last section.

2. New solutions for resource and service management

2.1 Approaches to resource and service management

ARCADIA

ARCADIA project is to provide a novel reconfigurable-by-design Highly Distributed Applications (HDAs) development paradigm over programmable infrastructure. A HDA is defined as a distributed scalable structured system of software entities constructed to illustrate a network service when implemented to run over a cloud infrastructure. The development, configuration and operation of HDAs entail many and multi-level challenges relating to the entire lifecycle of a highly distributed application. This lifecycle includes software engineering, optimization of deployment, infrastructural on-boarding, execution and the deprovisioning phases.

A HDA is constructed and deployed as a service chain of application tiers and virtualized network functions. **Service chain management** is performed through a HDA's entire lifetime. Deployment of application tiers is realized by the **creation of lightweight specialized unikernel virtual machines** (VMs). Such approach provides for high performance, high usage density of hypervisors; due to small footprint of unikernel VMs, and speed in performing management functions towards satisfying application's requirements as well as achieving providers' optimization objectives. **Multi-level monitoring** is performed to assure that: application's needs, service provider's policies and objectives and infrastructure provider's objectives are all satisfied at the same time. Monitoring is performed at the application component, the application service chain, the virtual execution environment and the physical host levels. A Rule based and a Complex Event Processing engines in cooperation with an Analytics, Optimization and Reasoning engine resolves conflicts in policies, produces predictive analytics, solves optimization problems and performs the necessary inference to trigger actions and execution of **devised algorithms towards meeting multi-objective goals** regarding HDAs deployment and operation with respect to service provider's and infrastructure provider's policies and objectives.

The project defines the management of an application from its creation from a developer to its operation over a programmable infrastructure. Annotated source code provides for embedding at the source code level multi-source requirements and driving operation of an application over a programmable infrastructure with **enhanced management capabilities** towards optimal deployment to satisfy the various involved stakeholders. Furthermore, **reusability and sharing** is promoted through providing a **component's' repository** for easily interfacing to a newly developed HDA already developed components, network functions, transformed legacy applications or multi-tenant services already in operation.

BEACON

The main goal of BEACON is to research and develop new innovative techniques to federate cloud network resources, and to derive the integrated management cloud layer that enables an efficient and secure deployment of federated cloud applications. The project will deliver a homogeneous virtualization layer on top of heterogeneous underlying physical networks, computing and storage infrastructures, providing enablement for automated federation of applications across different clouds and datacenters.

To implement these concepts, the BEACON project define a comprehensive framework for cloud federation that enables the deployment of different real use cases, and will provide a complete, ready-to-use framework implementation built upon open, interoperable, and standardized software to avoid vendor lock-in problems.

The new intercloud network capabilities are leveraged by existing open source cloud platforms, OpenNebula and OpenStack, to deploy multi-cloud applications. In particular, different aspects of the platforms are extended to accommodate the federated cloud networking features like multi-tenancy, federated orchestration of networking, compute and storage management and the placement and elasticity of the multi-cloud applications.

CloudLigthning

CloudLightning is proposing a **new cloud management and delivery model designed on the principles of self-organisation and self-management**. The promoted architecture is based on heterogeneous cloud resources to deliver services specified by the user using a bespoke service description language.

The self-organising behaviour built into, and exhibited by, the new cloud management and delivery model is the result in the **formation of a heterogeneous collection of homogeneous resource coalitions**, capable of meeting service needs. The quality, cost and performance of each heterogeneous service are expected to be different.

The primary use case for CloudLightning is **IaaS provision**. However, HPC use cases (namely genomics, oil and case exploration, and ray tracing) have been identified as suitable to validate CloudLightning, taking into account that the HPC Cloud is currently the most distant model from the Autonomic Cloud. The targeted IaaS provision may include public cloud service providers and/or companies providing, configuring or using private and hybrid clouds. In each case, CloudLightning resource management system is anticipated to offer better performance and greater energy efficiency.

CloudWave

DevOps describes the convergence of application development and operation activities. In a DevOps team, software developers and system administrators collaborate in joint task forces and work towards common goals. The vision of the CloudWave project is that a full-stack DevOps approach to cloud management can lead to more efficient usage of clouds as well as to better applications. This is achieved by aligning the goals of cloud application developers and cloud operators, and by allowing developers to leverage deeper knowledge of the cloud hosting environment. For cloud operators, the CloudWave model enables more efficient instance management, as application developers collaborate with the cloud provider, for example by exposing adaptation enactment points or emitting relevant business metrics. In return, cloud application developers gain deep insight into the internals of the cloud system, and can hence build and tune their application based on real-time feedback from the cloud. Similar to DevOps, the collaborative model of CloudWave removes friction between cloud operators and software developers by breaking up the black boxes that clouds and applications traditionally are to each other. CloudWave provides a reference implementation of these ideas based on Openstack.

ClouT

ClouT's main goal is to bring together the state of the art in Internet of Things and Cloud technologies. The aim is to enable smart cities to adopt an efficient and cohesive solution that includes infrastructures, services, tools and applications that will be reused by different city stakeholders such as municipalities, citizens, service developers and application integrators, in order to create, deploy and manage user-centric applications taking benefit of the latest advancements in internet of things and cloud domains.

ClouT's architecture elaborates on NIST's cloud reference architecture and defines three abstraction layers: City Infrastructure as a Service (C-IaaS), City Platform as a Service (C-PaaS) and City Software as a Service (C-SaaS). The functionalities of each layer, in the context of smart cities and IoT, are mapped on the functionalities of the corresponding cloud layers. In particular the C-IaaS layer includes functionalities to manage devices (sensors and actuators), IoT and not-IoT and related data, including storage; the C-PaaS layer provides the basic functionalities to process and access data and to compose services. C-SaaS layer is not strictly considered part of ClouT architecture because it is composed by the applications developed by municipalities and users on top of the other two layers. In this architecture pure Cloud functionalities become of fundamental importance to provide scalability, elasticity and resiliency to face the overwhelming amount of data expected to be produced everyday by a typical smart city IoT network. Furthermore Cloud infrastructure offers the possibility to quickly develop and deploy new services and functionalities to easier face requests from municipalities and citizens.

DICE

The main goal of DICE is to define an **model-driven engineering approach and a quality assurance tool chain to continuously enhance data-intensive cloud applications** with the goal of optimizing their service level.

The goal is to find architectural improvements to **optimise costs and quality** in Cloud-based Big Data applications. An decomposition-based analysis approach, where compute and memory requirements are analyzed and optimized separately, and simulation-based evaluation of performance metrics are used in optimization program. Fluid approximation of stochastic model is used to enable the simulating the behaviour of a system. Verification tools are used to assess safety risks in Big Data applications.

ENTICE

The ENTICE approach to service and resource management is focused on transparent, decentralised, and highly optimised management of VM operations in federated Cloud infrastructures. More specifically, the project focuses on the following VM management issues minimally addressed in the past.

Creation of lightweight VM images through functional descriptions. The project aims to support users with no expertise in VM image creation by delegating the optimisation of VM management operations to the ENTICE environment.

Distributed lightweight VM image storage. ENTICE delivers technologies that decompose user VM images into smaller reusable parts bringing a twofold benefit:

- it reduces the storage space by storing the common parts of multiple image only once;
- it will lowers the costs by ensuring that users only pay for the VM image parts that they cannot reuse from past images.

Autonomous multi-objective repository optimisation. ENTICE researches heuristics for multi-objective distribution and placement of VM images across a decentralised repository that optimises multiple conflicting objectives including performance-related goals (e.g. VM deployment and instantiation overheads, data communication, application QoS metrics), operational costs, and storage space. Through these heuristics, ENTICE ensures that commonly used VM image parts (e.g. just enough OS) are replicated and stored more widely. Thus, upon creating a new VM, common parts can be discovered and delivered from local repositories (i.e. of the provider who will host the new VM), while user-provided parts come from a different location (fostering cross-Cloud migration).

Elastic resource provisioning. The ultimate aim of this project is to use the optimised and lightweight VM management methods researched in the previous objectives to improve the elasticity for on-demand scaling of industrial and business applications in Clouds in response to their fluctuating compute and storage requirements.

Information infrastructure for strategic and dynamic reasoning. To support the optimised VM creation in the distributed repository, ENTICE develops a knowledge model of all entities and relationships for Cloud applications, including functional and non-functional properties of their underlying software components, QoS metrics, OS, VM type, and federated Cloud (e.g. SLAs), supported by strategic and dynamic reasoning. Strategic reasoning supports automatic VM packaging of applications based on criteria such as QoS functional properties, execution time, costs, and storage. Dynamic reasoning supports proper VM packaging and preparation based on dynamic (benchmark) information about the underlying federated Cloud (e.g. resource and network characteristics).

INPUT

The INPUT project is focusing on the design and the development of a novel platform for enabling and fully integrating **fog-computing services** into carrier-grade next generation Telecom Operator networks, powered by **Network Function Virtualisation (NFV)** and **Software Driven Networking (SDN)** technologies.

The INPUT framework will achieve the flexibility needed to provide cloud services to end-users, and resources to service providers, with innovative models beyond the classical ones (IaaS, PaaS, and SaaS). Orchestration and Consolidation criteria will dynamically optimize the **energy consumption** of the infrastructure, while meeting the users' QoE.

Moreover, the INPUT Project will provide a proof of concept illustrating the adoption of federated cloud computing and storage capabilities of network appliances to allow users to create/manage **private(/personal) clouds "in the network"**.

In order to minimize the OPEX and increase the sustainability of its programmable network infrastructure, the Telecom Infrastructure Provider will make use of advanced Consolidation criteria that will allow virtualized service instances to be dynamically allocated and seamlessly migrated/split/joined on a subset of the available hardware resources. This modularity will allow performing a more fine-grained management of the computing and storage resources and of their related **QoS/QoE** according to the user location, and enabling cloud service providers to easily upgrade/extend their offered functionality.

Mobile Cloud Networking

In today's situation, there are key challenges for telecom industries. The two key questions of the telecom and mobile industries are namely:

1. How can they optimise their CAPEX/OPEX[1,2], offering the same service at a lower cost or with greater profit and,
2. How can they incentivise their existing subscriber base to use new and innovative services by efficiently leveraging the vast amounts of infrastructure at their disposal and in doing so create new revenue streams?

These questions are addressed by the Mobile Cloud Networking (MCN) project. MCN is further focused upon two key ideas as a means to address these challenges.

1. The first idea is to exploit cloud computing as infrastructure for future mobile network deployment and operation. At the core, MCN is about the movement from systems that are self-hosted, self-managed, self-maintained, on-premise designed and operated, to cloud-native-based software design that respects existing standards and management and operation of those designs instantiated as services or indeed, network function services.
2. The second idea is more visionary and forward looking. It illustrates how we envision a future MCN service provider that fully endorses and adopts cloud computing services. These providers will leverage

[1] GSMA, "The Mobile Economy 2013", 2013 http://www.gsmamobileeconomy.com/GSMA%20Mobile%20Economy%202013.pdf
[2] PricewaterhouseCoopers, "We need to talk about Capex; Benchmarking best practice in telecom capital allocation", 2012

these services by building value-added services to commercially exploit as new innovative services, both traditional telecom services as well as new composed, end-to-end (E2E) services.

MCN is driven by cloudification and delivery of cloud-native applications and services with an aim of remaining compatible with ETSI NFV architectures. Nonetheless the considered sectors of MCN are not just IT but also mobile telecom. This is one of the **core propositions of MCN**, to use the cloud and to extend cloud services from the IT world into the mobile telecom world.

MCN and its architecture encompasses and integrates (converges) the three major stakeholder domains within this project. These three domains are:

1. **Radio Access Network (RAN)**: subscriber's wireless access is serviced. The challenge here within MCN is to virtualise the base station.
2. **Mobile Core**: subscriber's access through to internet services is given. The challenge here is the servitization and cloudification of the evolved packet core (EPC) architecture.
3. **Data Centre:** further services such as compute and storage (block storage, object storage, database etc.) services are offered. The challenge here is how these can be used efficiently within mobile and telecom environments.

In order to deliver such cloudified services MCN provides resource management framework independent delivery of resources. It accomplishes this through an architectural entity known as the Cloud Controller (CC). Through this, PaaS type resources (e.g. containers) can be delivered. IaaS resources are also delivered through using infrastructure template graphs, which are implemented using the Heat orchestration Template schema. This is achieved across CloudStack, Joyent Triton and OpenStack (nova/neutron) platforms.

To ensure resource and service selection, placement algorithms based on linear programming approaches are used. To ensure continued service delivery the CC also provides a means to monitor and notify resource owners when certain rules are breached. This is accomplished by building upon Monasca, which includes technologies of Zookeeper, Kafka and Storm.

In order to manage services offered from the MCN plattform (over 15 types), the OCCI[3] core model[4] is used to provide interoperability between MCN services. It is also between key entities of the MCN architecture. Each service type is instantiated through service managers. All service manager has an OCCI interface that allows the instantiation of a service type and supports all CRUD operations against that service instance.

MODAClouds

Within the MODAClouds approach we have experimented with model-driven development in the context of Multi-Clouds, enhanced with the possibility of exploiting models not only as part of design but also as part of the runtime. In this case the system model becomes a live object that evolves with the system itself and can be used to send back to the design time powerful information that enables a continuous improvement of the system[5] . This approach goes into the direction of offering a valid tool to support DevOps, that is, the ability to support development and operation in a seamless way.

In MODAClouds, Cloud services are represented and handled in a specific modelling language, namely MODACloudML. This is an extension of CloudML, which provides a domain-specific modelling language along with a run-time environment that facilitate the specification of provisioning, deployment, and adaptation concerns of multi-cloud systems at design-time and their enactment at run-time. MODACloudML is supported

[3] A. Edmonds, T. Metsch, A. Papaspyrou, and A. Richardson. Toward an open cloud standard. Internet Computing, IEEE, 16(4):15–25, July 2012.
[4] T. Metsch, A. Edmonds, Ralf Nyren, "Open Cloud Computing Interface - Core", Open Grid Forum, 2010.
[5] D. Ardagna et al, MODACLOUDS: A Model-Driven Approach for the Design and Execution of Applications on Multiple Clouds, Procs. MISE 2012, 50-56

by the MODAClouds IDE, which provides the functional, operational and data modelling environments, as well as some modules enabling the analysis of non-functional characteristics of a multi-cloud. From the point of view of the user, the IDE is the main piece of software to be used at design time. It realizes the model-driven engineering approach proposed by the MODAClouds, and its main output is the set of models and artefacts required by the runtime components to deploy, monitor and adapt cloud-based applications. Furthermore, it provides a common access point for a set of tools (its core functionality), and a set of accessory functionalities to aid the use of the other tools. The IDE is composed of five tool sets: QoS Modelling and Analysis Tool, Functional modelling Tool, Decision Making Toolkit, Deployment and Provisioning Component interfaces, Data Mapping Component.

The Execution Platform handles the deployments and execution on Cloud environments (offering IaaS and PaaS), allowing the user to gather a unified experience regardless the selected Cloud environments. Its main components are: platform system, build from the services which are tightly related with the MODAClouds environment; infrastructure system, build from services which handle the management of the cloud service, potentially enhancing them, and providing critical services required by other components; coordination system, offering services like distributed communication, coordination, data storage.

RAPID

RAPID focuses on **mobile remote acceleration** providing **acceleration as a service**, taking advantage of high-performance accelerators and high-bandwidth networks. Employing the RAPID service, compute or storage intensive tasks are seamlessly offloaded from the low-power devices to more powerful infrastructures (which may provide access to some accelerators), supporting multiple virtual CPUs and GPUs. RAPID offers a secure unified model where almost any device or infrastructure, ranging from smartphones, notebooks, laptops and desktops to private and public clouds can operate as an accelerated entity and/or as an accelerator serving other less powerful devices in a secure way. In this way RAPID provides CPU and GPU sharing between peer-to-peer devices as well as remote cloud offloading.

RAPID service is accompanied by a directory server providing the central management of the RAPID infrastructure. RAPID clients and servers are registered to a directory server, providing some information about available resources and required ones.

Each accelerated client can use the server to automatically find and connect to nearby accelerators and infrastructures with the required resources. Remote resources providers are assigned to each client at registration time (because of efficiency purposes) depending on the matching between required and provided resources, as well as on the possibility to fulfill certain Quality of Service (QoS) aspects that are defined in the application as annotations (i.e. number of GPU cores, cost, location, availability, etc.) Next, a runtime system, running on each such accelerated client, takes into account several parameters such as the local status, the environmental conditions, the task requirements, and the status of the accelerators it is connected to in order to decide whether local tasks (or incoming tasks if the entity also acts as an accelerator) should be executed locally or remotely. Novel scheduling algorithms, admission control policies and Service Level Agreements are employed to serve multiple accelerated applications efficiently on heterogeneous cloud infrastructures.

RAPID is also defining an easy-to-use task-based programming model and a novel runtime to automatically offload and execute the tasks transparently to the programmer.

SPECS

The core idea of the SPECS framework is to offer **security-as-a-service, through an security SLA-based approach**[6]. SPECS framework affects the security of cloud services by means of the integration of proper

[6] M. Rak et al, Security as a Service Using an SLA-Based Approach via SPECS, Procs. CloudCom 2013, vol.2, no., 1-6

security services in the cloud services invocation chain, which is initiated by an end-user that aims at accessing target services offered by external cloud service provider and providing the capability he/she is interested in.

The SPECS services may affect the security of the target services in various ways. In the simplest approach, the SPECS services may help the end-user to select the right provider for the same cloud service (using SLAs as a means for verification). Alternatively, they may add specific features that do not alter the capability offered by the cloud service provider, but improve the security involved in accessing it (e.g., activate intrusion detection systems, encrypt communication channels, etc).

SWITCH

SWITCH addresses the urgent industrial need for tools to support the development, deployment, execution and control of time-critical applications on Cloud infrastructures. SWITCH aims to realise an application-infrastructure co-programming and control model, which allows developers to program the application logic while taking into consideration quality of service and experience (QoS and QoE), and the programmability and controllability of the Cloud environment. SWITCH will provide a software workbench and an infrastructure comprising three components: *SIDE (SWITCH Interactive Development Environment)*, *DRIP (Distributed Real-time Infrastructure Planner)* and *ASAP (Autonomous Self-Adaptation Platform)*. SIDE will help developers to design and specify their applications and their time-critical elasticity, load balancing, monitoring and adaptation requirements. DRIP will then help find and set up appropriate infrastructures upon which to deploy and execute applications. ASAP will monitor the run-time behaviour of the application and operate with the aid of a knowledge base in order to monitor the performance, QoS, and QoE delivered by the application and, where necessary, adapt configurations at runtime in order to accommodate fluctuating load and bandwidth requirements.

2.2 Technologies that were developed or are under development for resource and service management

ARCADIA

Challenges addressed include (i) the design of a novel software development paradigm that is going to facilitate software developers to develop applications that can be represented in the form of a service graph –consisted by a set of software components along with the dependencies and interconnections among them- that can be deployable and orchestratable over programmable infrastructure, (ii) the design and implementation of an orchestrator (called Smart Controller in ARCADIA) able to undertake the developed service graph and proceed to optimal deployment and orchestration of the corresponding service/application and (iii) the design and implementation of a policy management framework that supports the definition of high and low level policies on behalf of a Services Provider that can be associated with a service graph or a set of service graphs along with the definition of a set of metrics for policies management purposes.

The under development Smart Controller is the application's on-boarding utility which undertakes the tasks of i) translating deployment instances and annotations to optimal infrastructural configuration, ii) initializing the optimal configuration to the registered programmable resources, iii) supporting monitoring and analysis processes and iv) reacting pro-actively and reactively to the configuration plan based on the infrastructural state, the application's state and the applied policies.

BEACON

BEACON develops a cloud network federation framework integrated with OpenNebula and OpenStack technologies for cloud management. This framework allows cloud providers, who use either OpenNebula or

OpenStack, to form federations and share resources, and provides the cloud user with extended service definition capabilities that are currently not supported by the OpenNebula service template or OpenStack service manifest. In particular, service definition are extended in order to:

- specify required automated high availability across datacenters. New expressions will be defined to specify which components need to be replicated or how frequently replication must be done. High availability is only possible in a tightly coupled federation and will only be supported between clouds running the same technology for cloud management and managed by the same organization.
- specify datacenter location aware elasticity. New service definition language expressions will be defined to specify constraints on the geographic location of application deployments. These location constraints will be enforced on the initial deployment and the elasticity mechanism. Geographic support will be partial, and will aim to support the use cases within the project testbed.
- specify requirements on network functions for the application's virtual network. New expressions will be defined to specify which network functions should be applied to the application's network traffic, and in which order they need to be applied. Available network functions will be limited to the ones that are available in the project testbed.
- specify firewall template definitions to the cloud middleware and run vulnerability analysis on the Virtual Machines (VMs) of a cloud deployment. The service definition language will be extended to refer to firewall templates that need to be applied to an application that is deployed in the cloud federation. The use of a firewall template ensures that the firewall configuration data and rules are independent of any specific firewall technology or cloud provider. The same firewall template will thus be deployable on either an OpenNebula or OpenStack cloud.

The federation network model under development can be used in different cloud federation architectures such as peer cloud federation, hybrid cloud federation, and brokered cloud federation.

CloudLigthning

CloudLightning is delivering **a new delivery model for a self-organising self-managing heterogeneous cloud** as well as a **proof-of-concept implementation**. Performance metrics, such as reliability, cost and power consumption are used to compare the self-organising and self-managing approach to the current state of the art in cloud delivery.

A gateway service[7] is used to front-facing the management system and to allow an application developer to initiate the life-cycle of a cloud service by presenting a blueprint that represents a workflow of services composed to automate a task or business process. The getaway receives the resource deployment options enabling the selection and the automatic deployment. The options are provided by cells composed of resource fabric comprising heterogeneous computer, storage and network resources organized in coalitions. The latest are formed statically or dynamically to reflect self-optimisatiòon, self-protection and contextual behaviour. Managers for the same type of resources can cooperate to form coalitions to meet the objective of efficient power consumption based in their local knowledge of underlying resource utilisation through autonomously monitoring itself and the environment.

CloudWave

At the heart of the project, monitoring data is generated, analyzed, and aggregated on all levels of the cloud stack, i.e., on physical, virtual, network, and application level. This data is further enriched with information

[7] T. Lynn et al, CLOUDLIGHTNING: A Framework for a Self-Organising and Self-Managing Heterogeneous Cloud, Proceedings of CLOSER 2016, SciTePress, in print

coming from external sources.We refer to this monitoring approach as 3-D monitoring. 3-D monitoring fuels two primary use cases, coordinated adaptation and feedback-driven development.

Coordinated adaptation is to improve the quality of adaptation decisions, taken by the infrastructure and application, through reasoning on the global state of the cloud stack provided by 3-D monitoring. By taking decisions in a coordinated manner, more effective adaptations are taken and operated by different components. To this aim, adaptation models of the application (e.g., turning optional application features on or off based on load) and infrastructure (e.g., scaling up or out) are captured by Feature-based models and adaptation plans are derived by an intelligent engine.

Conversely, feedback-driven development (FDD) aims to bring 3-D monitoring data to software developers, giving them a better understanding of how the application is actually operated (and adapted) at runtime. This goes way beyond traditional application performance monitoring (APM) solutions, as the CloudWave monitoring solution integrates data from the application stack with infrastructure metrics, information on triggered adaptations, and data from other applications launched by the same tenant. CloudWave demonstrates how this data can provide added value to software developers, for instance via visualizing (and warning about) performance-critical code directly in the Integrated Development Environment (IDE), or by enabling what-if analysis of performance and costs for different deployment options.

ClouT

ClouT effort in improving the state of the art in cloud technologies has seen specific development on the cloud storage interface, dynamic scalability and service development. ClouT's cloud storage architecture is composed of three main components: a Cloud Data Management Interface implementation known as CDMI broker, which is compliant with SNIA CDMI specification, an Hypertable/Hadoop cluster to store sensor data and a Swift Object Storage cluster to store binary data. The CDMI broker embodies the will to offer a RESTful interface specification compliant to services that consume cloud storage. On the backend side of things both Hadoop and Swift offer scalability right out of the box, but there's no mechanism for dynamic provisioning and instantiation of nodes; ClouT offers such mechanism via a specifically developed deamon that provisions and, eventually, de-provisions storage nodes when there is such a need. Therefore ClouT implements not only a readily scalable and replicated cloud storage behind a cloud-specific interface, but adds the elasticity required to handle critical events that may generate a spike in data storage requirements. Finally ClouT provides a Service Mashup Tool to enable users to easily deploy new services leveraging existing services provided by underlying layers and IoT devices. Service Mashup Tool is the ideal bridge between City Platform and Software as a Service layer making easier and quick the provisioning of new applications and services by a simple Graphical User Interface called Service Mashup Editor.

DICE

DICE is delivering a comprehensive quality-aware model-driven-engineering approach for Big Data applications. Its main tools will be[8]:

1. **DICE profile,** a novel data-aware UML profile to develop data-intensive cloud applications and annotate the design models with quality requirements.
2. **DICE IDE**, an Integrated development environment with code generation to accelerate development.
3. **Quality analysis**, a tool chain to support quality-related decision-making composed by simulation, verification and optimization tools.
4. **Iterative quality enhancement**, a set of tools and methods for iterative design refinement through feedback analysis of monitoring data.

[8] G. Casale et al, DICE: Quality-Driven Development of Data-Intensive Cloud Applications, Procs.MiSE 2015, 78-83.

5. **Deployment and testing,** a set of tools to accelerate deployment and testing of data-intensive applications on private and public clouds.

DICE tools rely on UML meta-models annotated with information about data, data processing and data transfers. The quality-aware tool chain is focused on the following quality dimensions:

- *Data-aware reliability:* quantitative annotations on the availability of a data source and on the reliability of data transfer and data processing operations.
- *Data-aware efficiency in resource utilization*: data operations consume resources such as memory, network bandwidth, storage and cache space, CPU time, software pool resources, etc. Service requirements at these resources are expressible through rates and latencies. Annotations give the ability to express service level objectives, such as maximum acceptable resource utilization levels.
- *Data-aware efficiency in costs:* a novel annotation to relate deployment and data transfer characteristics with costs. An example is the quantification of network costs, which can vary if a stream transports some data between components operating on the same cloud or between a component and an end-user external to the cloud.
- *Data-aware safety*: annotations dealing with constraints that must be guaranteed for safety reasons; specification of portions of the application that must logically and physically be isolated, together with the elicitation of formally correct requirements for a sequence of messages, states or events that relate application components.

ENTICE

Lightweight VM image creation. Existing methods for VM image creation do not provide optimisation features other than dependency management based on predefined dependency trees produced by third-party software maintainers. If a complex software is not annotated with dependency information, it requires manual dependency analysis upon VM image creation based on worst case assumptions. Existing research mostly focuses on preoptimising algorithms, not applicable to already available VM images. ENTICE extends preoptimising approaches so that image dependency descriptions are mostly automatically generated. The project also introduces new comprehensive postoptimising algorithms so that existing VM images can be automatically adapted to dynamic Cloud environments.

Distributed lightweight VM image storage. Today, commercial providers rarely expose the technologies they use to store the images for their users. Some providers allow sharing user-created images in a rudimentary way, e.g. using Cloud storage systems like S3 in case of Amazon EC2. Open source systems seldom allow the involvement of third-party repository services, but provide some simple centralised proprietary service instead. The limited sharing capabilities lead to non-optimal movement of VM images resulting in a slow Cloud VM instantiation across Clouds. Existing research in the area of image storage management targets only single Cloud environments. ENTICE researches and develops reliable and efficient VM image management method that reduces transfer costs, fine-tuned to federation-wide scenarios.

Autonomous multi-objective repository optimisation. Existing research is mostly limited to two objectives and employs a-priori methods that do not approximate the Pareto front, but aim to find a single compromise solution through aggregation or constrained programming. Moreover, most existing works target scheduling of applications to resources and do not target optimised placement of VM fragments in a repository distributed across several Cloud sites. ENTICE is the first project to research a truly multi-objective approach able to optimise the set of tradeoff solutions with respect to distributed VM placement in Cloud systems and involving three conflicting objectives: time, cost and storage.

Elastic resource provisioning. In modern Cloud environment, an important emerging metric is the elasticity of an application and its capability to adapt to different resource sizes and configurations, requiring advanced multi-experimental optimisation and analysis techniques. To this date, there is little understanding of the term elasticity in Cloud computing and tool support to improve applications and tune middleware infrastructures towards this metric. ENTICE will formally introduce a definition of an elasticity metric for Cloud applications and infrastructures and use it as a benchmark in quantifying and improving their on-demand performance.

Information infrastructure for strategic and dynamic reasoning. While Cloud computing offers clear advantages to applications, the semantic barrier for application developers and administrators of Cloud infrastructures is still enormous. Following the approach of Semantic Web, various researchers and projects tried to address this complexity. ENTICE extends the SLA@SOI results with a new concept of Pareto SLA capable of modelling the complete front of tradeoff solutions based on which further decision making processes can guide the (negotiation-based) selection of the "best" solution.

INPUT

The INPUT Project is fully committed to develop technologies and solutions to improve resource and service management in the edge network for hosting both software/network virtualised operations and fog applications/services. To this goal, in order to provide a thorough and complete coverage of the physical and logical components in the network, and to respect the related time constraints, two main building blocks with different targets are in charge of the programmable computing and network infrastructure, namely the Network and Service Management (NS-MAN) and the Network and Service Operating System (NS-OS).

The NS-MAN is responsible for the **long-term configuration** of the network, the administrative configuration of the infrastructure, the overlaying c**loud services and personal networks**, and for the **monitoring of the resources utilization and power consumption** of the INPUT infrastructure.

Differently, the NS-OS drives the **real-time configuration** of the programmable resources and the dynamic instantiation and migration of the virtualized cloud services according to users' locations, by performing the following three main tasks: **consolidation** (calculation of the optimal re-configuration of the infrastructure), **orchestration** (instantiation/migration of virtualized cloud services to the subset of hardware resources identified by consolidation) and **monitoring** (collection of performance measurements and alerts).

Mobile Cloud Networking

Mobile Cloud Networking's key outputs are the following:

1. Business scenarios and models for the delivery of cloudified composed services. This considers the cross-domain issues of offering composed and value-added services from the business perspective. The results of this are available in MCN's deliverable D2.1[9].
2. Mobile Cloud Networking Architecture. This architectural work is based in the principles of Service-Oriented Architecture and is applied to the combined domains of both Cloud Computing and Telecoms. Within the architecture there are three key entities, the Service Manager, the Service orchestrator and Cloud Controller. All of these cooperate through a common shared model, based on OCCI and described through acyclic graphs, the Infrastructure Template Graph and Service Template Graph. The results of this are available in MCN's deliverable D2.5[10].
3. Deliver of implementations of all services within MCN (15+) and a common orchestration platform that implements the architecture and service management lifecycle. These service implementations are available at MCN's github[11] and the orchestration platform, Hurtle[12] is available also at its website.

---

[9] http://www.mobile-cloud-networking.eu/site/index.php?process=download&id=123&code=ccaaad0c4777479f316069245856a8817f136d69
[10] http://www.mobile-cloud-networking.eu/site/index.php?process=download&id=263&code=aa37ba15e0479f1ecb7a696876ab498d7f3ff0ef
[11] https://github.com/MobileCloudNetworking/
[12] http://hurtle.it

4. The delivery of a comprehensive performance evaluation of delivering composed value-added services through geographically distributed service providers using the common orchestration platform, Hurtle. The results of this are yet to be released via MCN's website[13] as deliverable D6.4.

MODAClouds

MODAClouds model-driven approach for managing Multi-Clouds resources is supported by the MODAClouds Toolbox (available as open source[14]), comprised of the following elements:

1. Creator 4Clouds, an **Integrated Development Environment (IDE)** for high-level application design;
2. Venues 4Clouds, a **decision support system** that helps decision makers to select the best execution venue for cloud applications;
3. Energizer 4Clouds, a **Multi-Cloud Run-time Environment** providing automatic deployment and execution of applications with guaranteed QoS on Multi-Clouds.

Creator 4Clouds includes plugins focusing on analysing the QoS/cost trade-offs of various possible application configurations (Space 4CloudsDev), mapping high level data models into less expressive but more scalable NoSQL, deploying the resulting application on multi-cloud by exploiting the CloudML language. It is a unique tool supporting design, development, deployment and resource provisioning for multi-cloud applications that limits lock-in and provides features to assess the QoS guarantees required by the application. Moreover, it offers support to the definition of the application SLA. Note that CloudML is the only approach at date that supports seamless deployment and resource provision on both IaaS and PaaS. Moreover, there is no other tool than Space4Clouds that supports design time QoS analysis on Cloud deployments architectures in a multi-cloud context.

Venues4Clouds is the first attempt the problem of acquiring information concerning characteristics of cloud services by combining manual and automatic acquisition of data. In the context that the decision support for cloud selection has been based so far exclusively on cost and SLAs and not driven by risks and business perspective, the tool is the first attempt to offer a comprehensive approach to support users in identifying the right cloud services through the definition of a proper risk model.

Energizer 4Clouds includes the frameworks to support monitoring (Tower 4Clouds) and self-adaptation (Space 4CloudsOps), together with utilities that perform ancillary tasks in the platform (ADDapters 4Clouds). Energizer 4Clouds is one of the few approaches that addresses, in a single framework, the needs of operators willing to run their applications in a multi-cloud environment. The adoption of models to represent, analyze and evolve the status of applications is nowadays a hot topic in research; Space 4CloudsOps is the first one that enables an integration of this concept with a powerful language for deployment (CloudML). Through Tower 4Clouds, operators are able to perform complex monitoring and data analyses from multiple sources; while there are many monitoring platforms available in the literature and as commercial products, this tool is one of the first able to manage monitoring in a multi-cloud context. Moreover, it is empowered by a rich and expressive language for monitoring rules that allows application designers and operators to easily express their monitoring requirements and see them translated into concrete actions. Moreover, it is able to adapt itself depending on the evolution of the monitored application; thanks to Space 4Clouds for Ops, it identifies and actuates proper self-adaptation actions that take into account the current and foreseen state of the system under control.

[13] http://www.mobile-cloud-networking.eu/site/
[14] http://www.modaclouds.eu/software

RAPID

The RAPID framework, still under development, is composed of five basic components:

- RAPID Acceleration Client (AC), responsible for enacting task offloading;
- RAPID Acceleration Service (AS), which allows for remote execution of highly intensive tasks;
- RAPID Directory Service (DS), which provides the list of available physical machines in the cloud infrastructure to mobile clients or other components;
- RAPID Service Level Agreement (SLA) Manager;
- VM Manager, running at every physical machine acting as the responsible of creating, pausing, resuming and destroying VMs in the corresponding device or Cloud provider.

The **Acceleration Client** is composed of:

- The *Design Space Explorer (DSE)*, which decides whether a task should be executed locally or offloaded;
- The *Registration Manager (RM)*, which decided which RAPID node tasks will be offloaded to;
- The *Dispatch/Fetch Engine (DFE)*, which transfers the code to the selected Acceleration Server, receives the execution results and communicates them to the Application.

The **Acceleration Server** runs inside a VM that has the same operating system as the client associated to it. This entity is further composed of the following:

- The *Design Space Explorer*, which on the server side enacts the possibility of task forwarding to another AS, in case execution cannot be accomplished at the current one;
- The *Registration Manager*, which performs the registration process of Ass, and specifically the VMs hosting them, to DS;
- The *Dispatch/Fetch Engine*, which on the server side receives the code to be executed and then returns the results to the AC;
- *GPU-Bridger*, which takes care of communicating directly with the GPU hardware through gVirtus[15] in the RAPID case.

The **Directory Service** centralizes the knowledge of computational resources in the cloud infrastructure, providing information to Acceleration Clients about the available physical machines in the cloud or nearby devices whenever some computational resources could be acquired.

The **SLA Manager (SLAM)** ensures that the task execution on remote (cloud) resources will respect predefined levels of Quality of Service (QoS) and it also gathers information about the quality aspects that certain resources provider can offer when offloading.

The **VM Manager (VMM)** manages the computational resources of a physical machine, participating in the adequate management of available resources.

SPECS

The full SPECS framework is available as a prototype and released as open source in the SPECS official bitbucket repository[16]. The framework offers a set of reusable components for cloud application developers, in order to enable them to add SLA-oriented security mechanisms into their (existing) cloud services. The different

15 https://www.openhub.net/p/gvirtus
16 https://bitbucket.org/specs-team/

case studies and an available Portfolio, demonstrate that the proposed framework can be integrated "as-a-Service" into real-life cloud environments. The SPECS framework provides techniques and tools for:

- Enabling a user-centric **negotiation of security parameters in cloud SLAs**, along with a trade-off evaluation process among users and CSPs, in order to build and provide cloud services fulfilling a minimum required security level.
- **Monitoring in real-time the fulfillment of SLAs** agreed with one or more CSPs. SPECS monitoring services also enable notifying both users and CSPs when an SLA is not being fulfilled (e.g., due to a cyber-attack).
- **Enforcing agreed cloud SLAs,** in order to provide and keep a sustained security level that fulfills the specified security parameters. The SPECS enforcement framework will also "react and adapt" in real-time in case of possible alerts and/or violations, by applying the correct countermeasures (e.g., triggering a two-factor authentication mechanism).

SWITCH

SWITCH will be built using existing technologies and standards where available. The design of the workbench is such that individual subsystems can be replaced with alternative implementations where advantageous to do so, and to ensure that the key contributions of SWITCH survives the continuous evolution of virtual infrastructure technologies. Some candidate standards have been identified: for example TOSCA is to be used for describing applications for deployment onto virtual infrastructure, with other specific standards such as INDL to be used for describing planned and provisioned infrastructures.

There are also a number of specific technologies identified for prototyping SWITCH. For example SIDE will use the OSF (Open Science Framework) platform to manage application components, and use AngularJS to provide scalable APIs for interaction with back-end logic. The back-end of the Workbench will be a REST-based Web service implemented using the Django REST Framework. The serialisation of data transferred between the front and back-end of the workbench will be in JSON. DRIP will use a variant of the IC-PCP algorithm for infrastructure planning, while infrastructure provisioning will be based on technologies like Apache jclouds (for interacting with cloud providers) and OpenvSwitch (for implementing virtual switches in provisioning infrastructure). For execution, Apache ODE (Orchestration Director Engine) will be used for certain workflow applications. ASAP will be based on the JCatascopia monitoring platform. The Jena Fuseki triple store will be the basis for implementing the Knowledge Base. For the implementation of the time-series database (for continuous streamed data) Apache Cassandra will be used.

3.3 Expected impact on the service market of the action outputs

ARCADIA

Impact on SaaS providers. ARCADIA takes advantage of the reconfigurable by design development paradigm targeted at the design of highly distributed applications over programmable infrastructure to ensure the optimal use of the available resources based on the existing policies as well as the optimal runtime configuration in application and infrastructural level, based on orchestration provided by distributed collaboration of the Smart Controllers. ARCADIA's reference implementation and developed toolkits are going to facilitate application developers to design and develop infrastructural-agnostic applications and lead to the evolvement of novel and innovative paradigms for the deployment of advanced applications, boosting in this way the competitiveness of the software development industry.

Impact on IaaS providers. The deployment of highly distributed applications in ARCADIA will be realized over specialized and heterogeneous infrastructure and across a set of diverse networking domains, increasing in this way the complexity for the overall management as well as the efficient execution of applications. In order to handle this issue, a set of autonomic configuration, optimization and self-healing mechanisms will be integrated within the ARCADIA's components for reducing the overall management overhead. Furthermore, novel software technologies that tailor applications to the underlying hardware at deployment and run time will be developed. Dynamic re-configuration of the reserved infrastructure as well as application level runtime parameters will be supported. High performance and scalability issues will be also considered through distributed intelligence mechanisms based on the cooperation among the Smart Controllers, ensuring the high availability of the provided applications. The deployed developer ecosystems and the empowerment of all kind of heterogeneous resources will enable innovation, invention, creation and deployment of new business models.

BEACON

BEACON technology eases the deployment of multi-cloud applications, increasing the competitiveness of SMEs in IT, datacenters and cloud providers because it can increase their computing and storage capacity on an on-demand basis at a reduced cost. Thanks to open-source, Cloud Providers can easily test and evaluate new innovative advanced mechanisms, before offering them. They also benefit from open source by avoiding technology vendor lock-in and enabling software customization and integration.

Impact on SaaS providers: BEACON innovations will be of interest to those who use different IaaS providers and wish to federate heterogeneous (either public, private, or a mix of both) cloud infrastructure, and so customers of IaaS providers are also potential end-users of the BEACON solution. For example, there are some kind of distributed services (e.g. distributed web servers, distributed databases, etc.) that could deploy various cooperative/synchronized servers in different regions to improve the proximity and reduce the latency with different endusers located around the world.

Impact on IaaS providers: By federating separate physical infrastructures, IaaS cloud service providers are able to offer customers the ability to select a desired geographical region. There are several reasons by which a customer may want to have the infrastructure they are utilising in a specific region. These may include a desire for proximity to the infrastructure, reduced costs, or available resources in a particular region. Currently, the regions which IaaS cloud service providers offer are usually isolated from other regions and there is no interaction between them. This is the case for increasing fault tolerance and stability. BEACON's work to enable federated cloud networking would allow for infrastructure in different regions to be interconnected and would support distributed services, meaning that remote connections would not need to be manually configured.

CloudLigthning

Impact on SaaS providers. An important objective in creating new cloud management and delivery model as proposed by CloudLightning is to remove the burden of low-level service provisioning, optimisation and orchestration from the cloud consumer and to vest them in the collective response of the individual resource elements comprising the cloud infrastructure.

Impact on IaaS providers. A related objective is to locate decisions pertaining to resource usage with the individual resource components, where optimal decisions can be made. Currently, successful cloud service delivery relies heavily on the over-provisioning of resources. CloudLightning is addressing this inefficient use of resources and tries to deliver savings to the cloud provider and the cloud consumer in terms of reduced power consumption and improved service delivery, with hyperscale systems particularly in mind. By exploiting heterogeneous computing technologies, significantly improved performance/ cost and performance/watt are anticipated but also enabling computation to be hosted at large-scale in the cloud, making large-scale compute-intensive application and by-products accessible and practical from a cost and time perspective for a wider group of stakeholders. In each use case, relatively small efficiency and accuracy gains can result in competitive advantage for industry.

CloudWave

The main stakeholders in the CloudWave world are:

- Cloud Operators: The organization providing the Cloud (IaaS)
- Application Developers: The engineers who have developed the application (service) being hosted on the CloudWave cloud.
- Application Operators - The engineers concerned with improving the operation of the service.

Upon detecting the need for an adaptation action, the CloudWave Adaptation Engine attempts to devise an infrastructure adaptation strategy which will minimize resource costs to the Cloud Operator without impinging on SLA. This will translate into monetary savings for the Cloud Operator.

Application Developers are given access to CloudWave's suite of FDD tools, which utilize monitoring data to give developers insights into the behavior of their application which help them both improve the performance of their application, as well as rapidly created new features.

Application Operators are aided by CloudWave, both through FDD tools which analyze root cause for adaptation actions, as well being able to adjust the weights of the sub-adaptation engines which compose a prioritized list of potential adaptation actions.

ClouT

ClouT is a platform that integrates Cloud and IoT technologies and provides its users with a set of tools to develop services for citizens and cities. ClouT platform main feature is related to its capacity to captures, circulates and processes on the fly runtime but also historical data coming from different sources. Data is one of the most critical elements that will underpin the success of a city's transformation into a Smart City. Data generation continues to accelerate at an unprecedented rate, with the amount of data in the world expected to reach 44 zettabytes by 2020[17]. Service developers using ClouT will be able to develop services that were not possible before or would have foreseen high development costs or learning curbs because of the unavailability of data or the need to learn specificities related to the IoT technology. In ClouT the data is just ready to be used and tools are available to support data aggregation and service composition.

If we look at the main components that make up the ClouT platform we see that they also have individually an impact on the market, here are some examples:

- Sensinact Gateway: provides a unified access to a set of heterogeneous IoT devices. Developers do not need to know about the specificities. The gateway can run in the cloud to manage a very large number of devices and platforms.
- WEB Sensorizer: enables users to change static web information as active modern sensor data which with the Cloud-integration will provide scalability and dependability.
- Service mashup tool: enable users, with IT low-skills, to easily create mash-up of web services and open data, and to build complex user interface using a simple web dashboard.
- CDMI based Cloud: ClouT implements not only a readily scalable and replicated cloud storage behind a cloud-specific interface, but adds the elasticity required to handle critical events that may generate a spike in data storage requirements.

[17] http://www.emc.com/leadership/digital-universe/2014iview/executive-summary.htm

DICE

DICE delivers innovative development methods and tools to strengthen the competitiveness of small and medium independent software vendors in the market of business-critical data-intensive applications. Leveraging the DevOps paradigm and the innovative Big Data technologies of nowadays, DICE offer a complete open source solution for the quality-driven development of cloud-based data-intensive applications.

Impact on SaaS providers. DICE delivers innovative tools for optimizing the development and management of enterprise software applications. In addition to that, driven by software delivery models and new development methodologies, DICE produces an open source ecosystem capable of modernizing core enterprise applications by extending their capabilities. In particular, DICE helps the developer of a data-intensive applications in deciding deployment characteristics by identifying a deployment plan of minimum cost, subject to performance and reliability requirements.

Impact on IaaS providers. DICE monitoring platform streamlines the adoption of Big Data technologies, integrating monitoring services for multiple Big Data technologies, as well as system metrics (small footprint on monitored nodes). Monitoring platform enables cost reduction (open source), customization (existing technologies mixed together offering an advanced monitoring platform), plus a graphical user interface for displaying results and trends.

ENTICE

Impact on SaaS providers. ENTICE targets customers in the areas of customer relationship management, content management systems, enterprise resource planning, human resource and payroll, and supply chain management. Unfortunately, for this type of customers the existing SaaS deployment processes are complex and time consuming, which increases costs not only for the initial deployment, but also for the ongoing management and scale up of resources. The benefits of ENTICE will be the streamlined, automated and intuitive deployment process with simple, optimised, and predictable performance that helps to save time and effort to get customers cloud ready. Hence, the ENTICE environment will provide substantial advantages to many potential customers in SaaS domain such as lower capital expenditures, no need to manage upgrades and patches, and enterprise scalability.

Impact on IaaS providers. The ENTICE results will have a significant impact on the IaaS providers in terms of functionality and market footprint. By enabling Cloud-based network overlays as the vehicle of interactions between previously autonomous entities, the ability exists for companies with on-premise clusters to burst out to public Clouds in order to utilise on-demand resources only when needed and to meet capacity demands without having to overprovision hardware for the peak demand levels. On-demand images for this and other purposes can be made available with identical functionality, performance, and transparent operational costs (often unknown that keep end-users away from on-demand Cloud solutions), as an on-premise solution with reliable connectivity between ENTICE points of presence in multiple geographies across diverse Cloud providers. The ENTICE solution will significantly enhance Cloud IaaS provider functionality and can be considered as a value-add part of Cloud provider offerings by enabling a **VM repository-as-a-service** solution, improved import/export functionality and conformity integrated into Cloud provider software. An important effect of ENTICE will be the ability to process large data sets by expanding local capabilities to be subsidised by external resources as required, benefitting from improved efficiency, cost savings and QoS control by utilising a completely distributed and therefore theoretically infinitely scalable platform.

INPUT

INPUT will deliver a set of results constituting holistic solutions to **improve the capacity for designing and implementing resources' and energy usage efficiency within the Future Internet era**. The delivery of these innovative mechanisms will, on the one hand, facilitate their adoption by several Telecom Operators as well as software houses, as it is a cross domain solution, and, on the other hand, it will facilitate the promotion of additional **added value services**, build on top of these holistic solutions, to pave the way for innovative offerings from service providers.

To this goals, INPUT targets at the creation of an ecosystem that will facilitate the EU industry (including both Telecom Operators and Cloud Service Providers) to promote innovative ideas and technologies along with the preparation of the corresponding business plans. Based on the provided capabilities, a set of benefits is mainly recognized for significant reduction of their OPEX and CAPEX as well as the setup of a flexible services' provisioning model. In fact, the **Telecom Operator** is seen **as an innovator** that through the developed mechanisms has the potential to fully leverage and exploit its available resources in a flexible and energy efficient way. In more detail, through the **automated consolidation and orchestration** of the available resources, the operator is able to achieve their optimal use as well as proceed to sustainable infrastructural planning and evolution and thus achieve significantly reduced network CAPEX. Furthermore, through a set of autonomic networking/cloud capabilities, reduced OPEX will be also achieved.

Significant business value is also going to be created by the manufacturers through the **softwarization of network functions**, thus leading to cost reduction and providing the potential to develop and support novel mechanisms and, at the same time, push innovation in a quick way.

Mobile Cloud Networking

Impact on SaaS providers. Key shortcomings that were discovered and solved by the MCN consortium were how to:

1. deliver a service compositions across organizational borders. This is a particular thorny issue more so in the telecom industry, not no much an issue in the cloud computing domain. This can be seen as a need for greater control over the service delivered, especially with the traditional mindset related to QoS. The argument used here was that owners of a service instance should be given the capability to monitor their service and adjust it as far as a service provider was willing to offer. Related to this was proximity of the service instance to end users to which MCN investigated means to provide better placement of service instances based on geographic location (manifested as latency) and economic cost.
2. deliver services based on the business having existing software that they wished to offer as a service. This lead to the development of the Hurtle orchestration framework and a set of best practices in how how to design cloud native applications, especially for key telecoms platforms such as those delivering Radio Access Networks (RAN), the mobile core provider's network (evolved packet core, EPC) and the IP Multimedia Subsystem (IMS).
3. to verify and validate that all services in MCN could be composed and delivered through a per-tenant Radio and Core Network connectivity services. This was validated through two scenarios. The first delivered an over-the-top service through connectivity services, which was a Digital Signage Service. The second and more traditional was the delivery of an IMS service combining the connectivity services. Both had to respect end-user QoE and this was validated.

Impact on PaaS providers. The key impact here was the verification that network function virtualisation could be supplemented by new technologies such as OpenShift and CloudFoundry. This has lead on to further investigations on how key telecoms services such as EPC could be delivered using such technologies.

Impact on IaaS providers. CloudSigma was involved in the project, as well as a number of other OpenStack- and Triton-based testbed, and learnings from the project validated the claim that indeed basic cloud resources could be used to deliver both IT and Telecoms services, in combination, to a level that satisfies functional and nonfunctional requirements, including that of the expected end-user QoE.

MODAClouds

The main components of MODAClouds Toolbox are built with the idea to reduce the gap between development and operations teams of Multi-Clouds applications, according to DevOps philosophy. The toolbox helps lowering existing barriers between development and operation. Thanks to it, organizations of any size can build and run cloud applications driven by business and technical needs and quality requirements. Moreover, the design of the MODAClouds architecture included feed-back loop technologies, as the capability of the runtime to influence the design time is a very important feature to empower multi-cloud application developers.

Impact of SaaS. MODAClouds Multi-Cloud DevOps Toolbox, facilitates the development of Cloud applications and services by providing a way to extend current solutions to include the definition of business and functional requirements into application development. The toolbox offers various tools to put in place methodologies that help better collaboration among developers and operators, and therefore helps embracing DevOps movement.

Impact of IaaS. The Toolbox extends current deployment and runtime tools in order to consider multi-cloud environments and to assure automatic QoS awareness monitoring and responding to SLAs requirements and violations.

RAPID

Within the RAPID project the first public acceleration cloud service will become available and commercially exploitable. RAPID will directly impact the mobile and robotics market through its main concept: the integration of cloud computing, runtime systems, virtualization, real-time and monitoring in a performance and energy efficient way.

Impact on SaaS. RAPID platform introduces Acceleration as a Service that is a heterogeneous multi-level cloud-based service. More specifically, RAPID offers a programming model for application developers' that allows the easy and efficient use of cloud resources, permitting service providers to enter many other domains.

Impact on IaaS. RAPID offers advanced heterogeneous cloud infrastructures, which support virtualization with multiple CPUs and GPUs and advanced task scheduling algorithms to efficiently utilize cloud resources.

SPECS

The SPECS framework help Cloud customers to better understand security specification in SLAs in order to enhance the effective usage of Cloud computing. With SPECS, End-users can obtain a transparent view on the security levels agreed or expected from their Cloud services, thus rebalancing the current unequal relationship between End-users and service providers. From the End-user perspective, SPECS targets the creation of user-centric techniques and platforms to empower Cloud customers (in particular SMEs) providing a way to choose service providers via security SLAs. End-users will be provided with adequate support to make informed decisions regarding the trustworthiness of a service provider. As a direct consequence, it is expected that provided levels of trust and transparency will contribute to the industrial adoption of Cloud computing.

Impact of SaaS. SPECS platform enables the delivering of security services, described in details through Security SLAs. Cloud Service Customers (CSCs) are able to define at fine grain level the security features they need through a user-centric negotiation of Cloud SLA, that helps CSCs to negotiate Security Services offered by Cloud Service Providers (CSPs), by understanding the resulting trade-offs. In order to support CSCs to verify the correctness of the services offered by CSPs, SPECS offers innovative solutions for continuous security monitoring, which implements SLA monitoring solutions dedicated to continuously control the security mechanisms offered and to help ensuring the granted security service level objectives. Moreover, SPECS offers innovative Security Services to enforce SLA: when a cloud service does not grant the security features that CSC has expressed in the negotiated SLA, SPECS executes ad hoc security mechanisms that add and grant such specific feature.

Impact of IaaS. The SPECS framework, i.e. the software collection produced by the project in open source, can be used by Cloud Service Providers to integrate their service offerings with Security SLAs features and/or by developers in order to offer a Third Party solution that enhance the security of public Cloud Service Providers. In particular, the SPECS framework has been designed to empower Small and Medium-sized Enterprises with techniques and tools to improve their control and assurance over the security aspects of their Cloud services, taking into account the SLAs from the early definition of the supply chain used in the service provisioning. As a positive side effect, SPECS facilitates transparency between Cloud Service Providers and End-users to allow for a better assessment of the provided/requested security levels through SLAs. Indeed, SPECS is actively contributing to Cloud security SLA standardization initiatives.

SWITCH

The co-programming and control model used by SWITCH, and the software tools developed in the SWITCH project, will have impact on the lifecycle of time-critical application development, deployment and execution.

Impact on SaaS. The software methods and tools produced by SWITCH will take both the requirements of time-critical applications and the programmability of Clouds into account, accelerating the production of robust time-critical applications which can be offered to users on Clouds. The SWITCH Integrated Development Environment will integrate formal verification in a user-friendly environment in order to aid the programming of both application logic and runtime infrastructure in symbiosis, by referring to developer-defined quality constraints.

Impact on IaaS. The SWITCH project will deliver a time-critical Cloud application-oriented SLA model and negotiation mechanism as provided by the Dynamic Real-time Infrastructure Planner, allowing for a more nuanced business interaction between cloud providers and application developers. The cost efficiency of executing time-critical Cloud applications will be improved by allowing customising the runtime environment based on application requirements. The operational cost of managing time-critical application services on virtual infrastructure will be reduced by the Autonomous System Adaptation Platform provided by the SWITCH environment.

4. Gap analysis

ARCADIA

ARCADIA will provide a Holistic Novel Approach for the development of Reconfigurable- By-Design Highly Distributed Applications (HDAs). The envisioned Framework entails many innovative aspects, as far as the entire lifecycle management of HDAs is concerned. Thus; radical innovation aspects are introduced to Application Development phase, to Application on-Boarding phase and to Application execution phase. Regarding Application Development, ARCADIA Framework will use extensibility mechanisms that are provided by many third-generation- programming languages (e.g. JAVA) in order to create an Annotation Toolkit. This Toolkit will allow an HDA developer to provide source-code-level annotations that will refer to re-configurability aspects (e.g. balancing policies, QoS even Networking-level security). Although programming- language extensibility mechanisms are extensively used (the last four years) by many widely-adopted toolkits (Spring, Hibernate, JEE etc) there is no prior-paradigm of an extensibility mechanism that targets Infrastructural Reconfiguration.

BEACON

Most current cloud service providers offer basic network functionalities including network instantiation, network isolation, and address management. However, these network services are very limited and do not allow users to have advanced control over these virtual networks. The cloud network model should evolve to an on-demand provisioning model that, relying on the existing virtualization technologies, enables users to define customized virtual networks, to have advanced control over network configuration, and to implement advanced features (e.g. scalability, performance, QoS, security, and federated networking support). There are some basic network functionalities that are currently offered by most cloud infrastructures, such as the OpenNebula or OpenStack-based clouds, that allow users to instantiate virtual networks, to manage addresses, and to isolate traffic from different networks. However, these functionalities have to be extended with advanced network functionalities, in order to design challenging future cloud infrastructures, able to improve the network management both inside the cloud and across different clouds

CloudLigthning

Autonomic Clouds features are only partially available today. The main characteristics that are distinguishing an Autonomic Cloud from current Clouds are: follows a contextual behavior through methods of self-management, self-tuning, self-configuration, self-diagnosis, and self-healing; presents itself as a robust and fault tolerant system; it is easy to manage and operate the services and deployments by using techniques for the design, build, deployment and management of resources without human involvement. Such level of automation is not yet achieved and will not be achieved soon. This is clearly proved by the current approaches in HPC Cloud were bare metal and classical pre-allocation style are still dominating.

ClouT

The introduction of Cloud in IoT paradigm solves two main gaps in smart cities context:

1. Lack of elasticity and scalability "on demand", that force municipalities to overprovision their premises of useless hardware to face potential data or processing bursts increasing dramatically costs or to renounce to face them
2. Difficulties to introduce new services based on data collected from sensors or actions available by actuators because legacy development models are expensive and slow.

In its architecture, ClouT includes support for open source Infrastructure as a Service platforms, an elastic Platform as a Service, including Service Mashup Tool, making it easier to continuously develop cloud applications that scale dynamically according to the workload, Big Data technologies to analyze and make use of data collected and offers a dynamic, elastic, auto-provisioned cloud storage behind a standardized RESTful API. All of this while maintaining focus on bringing cloud advantages to smart cities.

DICE

As data-intensive applications run on complex massively parallel systems, **scalability issues** are often encountered by quality assurance tools or monitoring systems. Adaptive and fast reactive systems should be designed in the near future to cope with the fast development of Big Data technologies. Moreover, the Big Data technologies are currently evolving to allow innovative data mining methods to be applied and the Cloud-based services should keep track of these evolution.

Mobile Cloud Networking

Speed of service instantiation is greatly influenced by the particular cloud management framework and type of virtualisation. If virtualisation approaches are to be further placed into production they need to be fast and small. It is common for over one million packets per process to processed but in order to do so new approaches such as unikernels and libraryOS's are required. Another potential for addresses the high performance requirements of the telecoms sector is to invest more research into rackscale hardware architectures and the supporting resource management software required, however as with any new hardware platform the current costs associated make this cost-prohibitive especially when the cloud computing zeitgeist revolves around the idea of commodity hardware.

MODAClouds

Model-driven engineering techniques have proven to be useful for the management of Multi-Cloud resources. However the **automatic generation of the codes for Multi-Clouds** starting from abstract model is still an objective far to be achieved due to the diversity of the Cloud services and services that are currently available.

RAPID

Embedded systems are becoming more and more powerful. Mobile applications are becoming more and more performance and power hungry pushing the boundaries of devices' capabilities to the limits. The problem is becoming even more noticeable when applications are executed on older devices. Cloud computing and mobile connectivity seem to be an attractive solution to this challenge. However, the **overhead in energy and response time** is often greater than the offloading savings; this means that there still some work to do on decision making for **offloading computation tasks**, especially in the aspects of adequate resources allocation, QoS management and remote access to accelerators.

SPECS

Service Level Agreements are currently imposed by Cloud service providers. The user requirements are not necessarily catched well in these SLAs leading potentially to a waste of resources and dissatisfaction of the customers. Security SLAs are concrete examples of particular types of SLAs that are pushed by the user communities, standardization bodies and public institutions. Their **adoption along with other types of customized or purpose-oriented SLAs** can lead to a new level of quality of the Cloud services and their resource management systems.

SWITCH

Current time-critical application programming models lack consideration of the controllability of the infrastructure; they thus do not exploit the potential benefits offered by the programmable infrastructure provided by cloud environments. In particular, there is a lack of effective description mechanisms for specifying application logic, system quality constraints, and infrastructure controllability. Moreover, tools providing effective support for application-infrastructure co-programming have not yet been produced.

There have been many studies on the application of various optimisation mechanisms in selecting resources. However, there is currently no semantically well-modelled mapping between the application quality of user experiences, and infrastructure-level QoS attributes. This renders the modelling and negotiation of an SLA between application and the resource providers difficult at best. Although optimisation of network protocols in data communication has been extensively studied in the network and communication domain, the inclusion of network controllability (such as SDN technologies) in applications' data delivery services is still at a very early stage.

Although self-adaptive software has already been studied in software engineering, and autonomous systems, there lack effective mechanisms to semantically couple the various kinds of monitoring with application logic, and to make use of this information in order to adapt system-level performance by controlling both application and infrastructure. The existing self-adaptation intelligence focuses either on controlling application architecture *or* on the service quality of the runtime infrastructure; there is a lack of co-controlling mechanism addressing *both* aspects.

5. Conclusions

The current theoretical approaches and technological advances proposed for the resource and service management in Cloud environments by the cluster actions are expected to have a considerable imapct on the market of software developers and infrastructure service providers. We have provided in this paper a snapshot on the on-going activities as well as on the expectations of the action in what concerns the market influences. However the intensive activities have reveal also several gaps that are still not filled by the current theoretical or technical solutions and can be subject for future research and development actions.