Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 68 (2015) 227 – 237

Procedia Computer Science

HOLACONF - Cloud Forward: From Distributed to Complete Computing

Challenges emerging from future cloud application scenarios

Keith Jeferry[a]*, George Kousiouris[b], Dimosthenis Kyriazis[b], Jörn Altmann[c], Augusto Ciuffoletti[d], Ilias Maglogiannis[e], Paolo Nesi[f], Bojan Suzic[g], Zhiming Zhao[h]

[a]*euroCRIS, Anna van Saksenlaan 51, 2593 HW Den Haag, The Netherlands*
[b]*National Technical University of Athens, Heroon Polytechniou 9,15780 Zografou, Greece*
[c]*Seoul Nationa University, Gwanak-Ro 1, Seoul 151-742, South-Korea*
[d]*Department of Computer Science, University of Pisa, P.le B. Pontecorvo, Pisa I-56122, Italy*
[e]*Department of Digital Systems, University of Piraeus, Piraeus, Greece*
[f]*Department of Information Engineering, University of Florence, Florence,Italy*
[g]*Institute for Applied Information Processing and Communications, Inffeldgasse 16a, 8010 Graz, Austria*
[h]*University of Amsterdam, Science Park 904, Amsterdam, 1098XH, The Netherlands*

Abstract

The cloud computing paradigm encompasses several key differentiating elements and technologies, tackling a number of inefficiencies, limitations and problems that have been identified in the distributed and virtualized computing domain. Nonetheless, and as it is the case for all emerging technologies, their adoption led to the presentation of new challenges and new complexities. In this paper we present key application areas and capabilities of future scenarios, which are not tackled by current advancements and highlight specific requirements and goals for advancements in the cloud computing domain. We discuss these requirements and goals across different focus areas of cloud computing, ranging from cloud service and application integration, development environments and abstractions, to interoperability and relevant to it aspects such as legislation. The future application areas and their requirements are also mapped to the aforementioned areas in order to highlight their dependencies and potential for moving cloud technologies forward and contributing towards their wider adoption.

Peer-review under responsibility of Institute of Communication and Computer Systems.

Keywords: Cloud computing; future application scenarios; challenges; complete computing

* Corresponding author. Tel.: +44 7768 446088.
E-mail address: keith.jeffery@keithgjefferyconsultants.co.uk

Peer-review under responsibility of Institute of Communication and Computer Systems.
doi:10.1016/j.procs.2015.09.238

1. Introduction

The global penetration of the Internet, along with the affordability of Internet devices (laptops, tablets, smart phones, sensors, gaming consoles), formed the need for IT mobility that led to the emergence of Cloud[1]. In this context - at an incredible rate - cloud environments transform and revolutionize the way individuals, business and government are managing their services and their data accelerating innovation, improving time-to-market and offering added flexibility. A large number of applications from various domains including file hosting, social networking, office applications, business applications, games and numerous others have been ported from desktops or servers to cloud infrastructures. This provides clear benefits to users, who are released from the burden to install, configure and maintain applications, as well as hosting hardware[2]. The emergence of cloud concepts extended the system environments of enterprises beyond their organizational or physical boundaries, bringing a whole new range of usage and integration scenarios. In the cloud context, applications and data can be hosted, or exchanged through a wide range of disparate entities, while the services itself can be interconnected using heterogeneous interfaces and platforms.

However it is a fact that while cloud computing environments mature and become more ubiquitous they are mainly considered a "utility" for the aforementioned applications and for new, web-based application industries where "Born on the Cloud" is the usual delivery model. This is not the fact though for future applications and services, since they set a number of additional research and technological challenges for cloud computing, or in general the computing continuum, including (collaborative) Internet of Things (IoT), Cyber-Physical Systems (CPS), advanced networking infrastructures (e.g. 5G) and Cloud environments (platforms and infrastructures).

The aim of this paper is to investigate new technological challenges and goals for the evolution of Cloud services and environments. To perform this, initially a number of key application areas and capabilities are identified in Section 2, that are not met by current advancements, thus can generate requirements and goals for the progress of relevant Cloud technologies. These advancements are then highlighted in the remaining sections, at a section granularity. Finally the paper concludes in Section 7.

2. Future Application Scenarios

A number of application scenarios may be identified that their requirements cannot be met (or are met inefficiently / partially) by current cloud technologies but would be interesting to be investigated for future scenarios, since the requirements they pose will drive the evolution of cloud technologies. These scenarios may span from application synergies to new and improved applications running or exploiting cloud features. In the following sections we briefly introduce such scenarios along with the requirements and challenges they highlight for future cloud environments.

2.1. Joint Collaborative Business Intelligence platforms with multiple data sources

Although small and medium-sized enterprises (SMEs) are beneficial for an economy (e.g., employment of many people, production of innovative products), they lack resources to collect, analyze, and to utilize business data for future product developments. Thus, provisioning of Business Intelligence (BI) services for SMEs, that is already available as an offering, might run on different IT service platforms and therefore should be able to access data, software, and hardware resources on a per-use basis. One key powerful feature of such solutions would be to embrace the sharing of data with other SMEs, in order to build up a large database of suitable data for the analysis. For this, it would be helpful, if IT service platforms could provide an incentive structure for sharing data, which resolves their different preferences. Furthermore, the value exchange between the stakeholders (i.e., IT platform providers, SMEs, software service developers, and data providers) in the IT service platform ecosystem needs to be understood, in order to define the interfaces of the IT service platform. Therefore, the value exchange will have an impact on how data is managed and on how data can be shared.

Although the benefits of such business intelligence services and synergies are eminent for SMEs, their provisioning requires *interoperability between IT service platforms, at the software service and the data level.* Without this interoperability, enterprises might get locked in an IT service platform (i.e. service or data lock-in) and without being

able to access other (additional) services, will stay out of the market, or will not be offered comprehensive business intelligence services by software developers.

2.2. Knowledge on the Cloud

In the last years a significant convergence of knowledge, cloud and cultural heritage (information) has been observed, mainly by realizing that cultural heritage is knowledge and knowledge is becoming more accessible and easily managed into the cloud (e.g. in art modeling and media distribution[20] and the performing arts[21]). Furthermore, Open Data initiatives are more and more coming into light, for a variety of use cases, from environmental to research purposes, or matters of heritage or document preservation, for example through the creation of online repositories of digitalized material[28]. All this amount of "born digital" data raises the need for managing the corresponding information (and knowledge following data analytics) in a way ensuring the *exploitation of all available data sources - i.e. semantic interoperability of data*, thus allowing different applications to obtain the aforementioned information / knowledge.

2.3. Knowledge from the Cloud

The availability of distributed services and resources is reshaping the way in which people learn and teach, the way they "interact" with knowledge: for example the successful diffusion of Massive Open Online Courses (MOOCs), online tutorials, screen-casts and all sorts of online learning supports the statement. Not only the organizational aspects of educational institutions are under stress, as shown by the success of the Moodle project, but the tools themselves that the teacher uses to transfer their knowledge and competence to the students are under discussion. Such services are available in the Internet, and require a minimal preparation of the host. A challenging task like the administration of a Laboratory Classroom is thus simplified, and in many cases a Bring Your Own Device (BYOD) approach is feasible. Embracing a BYOD approach for the infrastructure of an educational institution, delivering additional resources in a private or public cloud, is a tempting option. But there are caveats the administrator should take into account, such as service costs, heterogeneous devices and variability of session or content size.

These features are common in cloud technologies, especially Software as a Service (SaaS) offerings, which are usually accessible through a web interface, using a browser. Cloud providers tend to offer differentiated products, both in terms of complexity and of cost, and service reliability is a key feature. SaaS offerings also tend to have easy interfaces, browser-based GUIs (solving the heterogeneous devices issues) and can bill based on user participation. Therefore it can be concluded that cloud technology has the potential to improve the quality of our teaching, reducing at the same time the investment in the computing infrastructure of educational institutions. It is possible to move from a traditional classroom to a virtual classroom that extends, and does not replace, the face to face experience of a traditional classroom. This teaching application (as a representative example of a knowledge acquisition scenario), highlights the need for future cloud technologies to facilitate *"understanding" of application characteristics and the generated knowledge in a dynamic way*, thus enabling the platform and infrastructure to provide the required resources that accommodate evolving (as knowledge is generated) requirements of the applications during runtime.

2.4. Software Development on the Cloud

Not surprisingly, the software development environment itself, which incorporates a wide tool-chain of editors, compilers, runtime environments, debugging and documentation tools etc., has also found its way to cloud infrastructures, which can offer shared programming and computing resources, together with powerful collaboration and code sharing facilities[14]. During the last few years, a large number of cloud Integrated Development Environments (cloud IDEs) have emerged, including Compilr (https://compilr.com), JS-Fiddle (https://jsfiddle.net), Cloud9 (https://c9.io), AkShell (https://github.com/akshell), Codenvy (https://codenvy.com), Eclipse Orion (http://www.eclipse. org/orion), Coderun (www.coderun.com), Kodingen (https://koding.com) and others. Typically, these platforms support code development with the aid of an enhanced web-based editor, while code compilation, execution and testing is performed on the remote cloud infrastructures. In several cases, some sort of code reuse through shared user repositories enhances code productivity, while some platforms provide a cloud-hosting

environment for post-production application execution. However, the increasing need for *rapid and cost effective software creation*, poses even tougher requirements to the *process of application development*. Collaboration among several developers and software components reuse are essential factors, in order to deliver high quality applications with reduced time to market.

Furthermore Model Driven Development (MDD)[16,17,18,19] is a successful software engineering process, which exists in traditional application development, and could assist cloud IDEs to fulfill their purpose. MDD decomposes system design and operational logic from implementation details, by utilizing appropriate abstractions expressed as models. This decomposition greatly simplifies the software development process and is able to automate substantial parts of it, by exploiting automatic mechanisms for code generation. Although typical cloud applications share a large number of common features and components, *MDD is not fully incorporated in modern cloud IDEs*.

2.5. Joint application and infrastructure controllability

A critical matter that needs to be addressed by modern application developers is the fact that with the advent of cloud technologies, elastic resources or services are available, however for them to be optimally utilized and efficient, their control needs to be tightly coupled with the deployed applications, in order to fully exploit this potential per case. More advanced resource management engines need to be incorporated at all levels of the cloud ecosystem, from hardware, to cloud management software, the network and up to the cloud applications themselves. All these layers should collaborate in an efficient way to save resources (a concern for cloud providers), without violating quality of service as expressed in the relevant Service Level Agreements - SLAs (a concern for cloud customers and developers). Thus, all cloud components including applications and application development environments will greatly benefit from intelligent management engines that are able to monitor resource consumption, analyze the current status, predict future demands, decide on more resource-efficient configurations and enforce/request those new configurations within the hosting environment. This tighter integration actually highlights that *applications cannot be agnostic of their future: the deployment, instantiation and operation in cloud environments and as a result the knowledge of the characteristics, particularities and constraints of such environments*. The description of a representative set of these applications follows.

2.5.1. QoS / QoE critical applications on the cloud

Many time-critical applications (applications that must respond immediately to time-sensitive events over a prolonged period) often have very high business value (e.g. on-demand business collaboration platforms) or social impact (e.g. disaster early warning systems). These applications demand a high standard of Quality of Service (QoS) (e.g. tsunami emergency response time) or Quality of Experience (QoE) (e.g. smooth delivery of ultra-high definition audio and video for live events), but are very difficult to develop and operate because of their distributed nature and the high requirements they impose on the runtime environment - in particular the sophisticated optimization mechanisms needed to develop and integrate system components that must interact seamlessly with one another. Cloud environments are capable of providing virtualized, elastic, controllable and high quality on-demand services for supporting these kinds of complex distributed applications. Indeed, many cloud providers already provide many of the technologies needed to develop and deploy these applications. However, what time critical applications still need from the cloud is the ability to control the selection and configuration of infrastructural components in response to changing requirements and environmental pressures. Unfortunately current cloud environments lack the tools and application programming interfaces that would allow the developers to exert such control on the underlying infrastructure in an intelligent, semi-autonomous manner.

Several general requirements can be extracted from the analysis of different time critical cloud application scenarios: (i) acceptable performance relies on the *satisfaction of high-level application-specific requirements*; (ii) there need to be *mechanisms to actually verify that applications can achieve the desired QoS in the first place*; (iii) the *integration complexity* of applications requires specialised knowledge of how to model and control software and infrastructure parameters in a cloud context; (iv) the use of virtualized resources requires expert insight into the *selection of appropriate cloud platforms and programming models*; (v) the configuration of the infrastructure to support the desired QoS requires a *standard, reusable interface* by which to tailor it; (vi) data-intensive communication within applications requires *virtualized storage and communication configurations based on*

functional descriptions that characterise data throughput and latency between application components; (vii) both applications and infrastructures need to be able to *adapt their configurations* in order to cope with quality-on-demand at run-time; and (viii) *dynamic Service Level Agreement (SLA) negotiation* with (federated) cloud providers is required when adapting such infrastructure.

2.5.2. Adaptable parametric applications

Applications now should become more intelligent, "live" and understanding of the environment in which they are executing, whether this is a Cloud environment or a limited resources Internet of Things (IoT) endpoint. They need to adapt accordingly and shift their parameters of operation in order to fit the specifics of the environment[15]. Thus evolvable and parametric implementations need to replace previously static developments in a more generic framework rather than following static analysis[7]. Based on the target deployment facility, the instance is able to understand the characteristics of the latter and configure the parameters accordingly. For example, the deployment on a limited hardware base will redirect intense computational work on cooperating Cloud resources, while trying to minimize energy consumption. This may be performed by investigating tradeoffs between data sampling and data transmission, thus calculating locally average values instead of sending every value. On the other hand, when deployed on Cloud infrastructures, cost minimization for the service usage may be the dominant goal.

For example[5], in order to exploit elasticity of Cloud services, Key Performance Indicators are used together with service offerings to enable this like Amazon Web Services (AWS) Auto Scaling. This has an effect of course on the overall cost for the owner of the Cloud-based application, but given that along with it the revenues from the usage increase, they are able to make more profit. The problem in this process is when an application that is available to external clients through the public domain becomes a target for Denial-of-Service (DoS) or Distributed (DDoS) attacks. In that case, the requests generated towards the server do not generate profit for the owner but due to the elasticity policies they increase the cost of deployment, thus leading them to monetary loss and eventually bankruptcy, also known as Economic Denial of Sustainability[6]. Thus suitable programming abstractions should be in place to monitor this tradeoff between elasticity and generated profit.

Furthermore, considering expectations of further developments and transitions towards cloud-based services, especially in the area of services similar to SaaS and PaaS offerings[23], the number of service interconnections and integrations is expected to grow rapidly. As these services are external to each other and do not reside inside the premises of one organization anymore, the *proper governance of related interconnections* should be assured. The transfer of data or service execution across diverse organizations requires the *clear definition and separation of each entity's capabilities, liabilities and obligations*. Furthermore, it requires a shift from predefined or hardcoded approaches to more interoperable and comprehensive security policies whose execution performs beyond the originating environment.

2.5.3. Generic application templates

In this case, the main advancement envisions the cloud services user that is able to select from a set of workflows or service interconnections, potentially based on specific application templates and their needs, having only to configure and deploy them with minor details (e.g. account credentials). This is tied to a specific application instance and relates to a specific application architecture that is widely accepted. This would have a similar effect to Wordpress for the building of web pages. Having templates of applications can significantly speed up the development cycle and enable non cloud experts to undertake faster the respective technologies. Thus, the main challenge / requirement actually refers to the provision of technologies that would enable applications to exploit in terms of *"ready to use templates" a set of different deployment patterns, instantiation and configuration strategies*.

3. Cloud Service and Application Integration

Following the aforementioned scenarios, a number of key advancements can be identified and further analyzed. Originating from the requirements of Sections 2.1 and 2.5, the need for tighter Cloud Service and Application Integration arises. With the introduction of cloud paradigm, typical service and process integration scenarios were extended with the following integration patterns: (i) cloud to on-premises; (ii) cloud to cloud; (iii) on-premises to on-premises; and (iv) eCommerce business-to-business integration[26].

The integration approach applied by the category of cloud integration services can be perceived through two primary integration models. The first approach to service integration is based on the concepts of composite cloud services and service brokerage. These are the services that derive the additional value for their core applications by integrating with one or more third-party cloud services. In this model, the cloud vendor enriches its primary service with the capabilities of services offered by the other vendors. This way, the cloud vendor generates additional value for the customer by integrating its primary service with the services provided by other companies, enabling the customers to *use, combine and integrate complementary products delivered by various organizations*.

The second paradigm that reflects integration models of cloud services is based on Integration Platform as a Service (IPaaS) approach. Instead to enrich the value chain of their core systems by integrating services of other vendors, the core business value of IPaaS is reflected in their ability to serve as an integration and automation platform by unifying, bridging and orchestrating various backend or frontend cloud services for their customers, directly in the cloud.

One of the significant improvements and optimizations that the integration platforms bring is the ability to *abstract and transparently handle the wide range of chained backend services and tasks*, supporting their further reuse and specialization. Integration platforms hence relieve their customers of the complex tasks that include administration of various APIs, platforms, systems development and lifecycle management.

3.1. Security and authorization aspects of integration platforms

Organizations that apply these models in their environments effectively extend or transfer the execution of their business processes in a cloud based, complex, multi-organizational and multi-layered setting. Besides providing non-disputed benefits, such setting additionally raises the new issues as well, especially in the domain of information security and data protection. The workflow of integration platforms encompasses the usage of organizational accounts at third party providers to perform predefined tasks or complex workflows. The usage often employs external web API (RESTful or SOA based) and requires the issuance of authorization tokens for platforms to access or consume the services. As the most of the solutions in the integration market rely on OAuth 2 based authorizations[25], they both inherit its advantages and drawbacks. OAuth 2 approach is relatively easy to integrate, but provides only limited set of features, focused on simplified authorization use case. The resulting integration therefore lacks fine-grained, policy-enhanced, assured and auditable data flow control and monitoring. Consequently, the organizations that consume composed cloud services, service brokerage or IPaaS systems are restricted in the degree of control or awareness over the flow of their data and usage of their services in distributed environments.

Being focused on integration platforms and SaaS integration, it can be furthermore noted that the similar issues apply to other web-scale based integrations and data sharing on the web. Although there exist other protocols such as UMA[24] or XACML[27], their adoption level across cloud providers is relatively low. Furthermore, the scope of their capabilities does not satisfy the requirements of chained, multi-level service integrations in the full extent and across multiple dimensions of security in cross-domain integrations. From this point, the lack of *advanced security mechanisms in integrated platforms* can rather be considered as an instantiation of a general problem that faces other areas of service integration.

3.2. Service Network Definitions and Semantic Descriptions

Advanced infrastructures potentially enable quality-guaranteed runtime environments, and we can see two important foci directly related to time critical applications specifically: (i) the transfer of High Performance Computing (HPC) environments to virtualized infrastructure such as the cloud, and (ii) the emergence of advanced network technologies—not only advanced hardware (e.g. quality-guaranteed optic networks) but also advanced protocols for controlling network behaviour and quality of service, such as Software Defined Networking (SDN)[8]. These advanced infrastructures provide developers opportunities to program and customize qualified runtime environments for time critical applications. However, to do this effectively also requires an effective planning model for defining the virtual runtime environment, advanced network services for optimised communication, and technologies for issuing SLAs and negotiating them in real time. We can identify a number of technical gaps that need to be addressed:

i. Current time critical application programming models lack consideration of the controllability of the infrastructure and 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 holistically*. Moreover, tools providing effective support for application-infrastructure co-programming have not yet been produced.

ii. There have been many studies on the application of various optimization mechanisms for selecting resources. However, there is currently *no semantically well-modelled mapping between the application-level quality of user experience, and infrastructure-level QoS attributes*. This renders the modelling and negotiation of an SLA between an application and the resource providers hosting the application difficult at best[9]. 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[10].

iii. Although self-adaptive software has already been studied in software engineering[13], and autonomous systems[11], 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[12]. 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.

An example of such description abilities lies on the knowledge acquisition (e.g. teaching) applications domain. A first step consists in understanding how resources that have been created for generic purposes, can be adapted to a teaching environment, thus encouraging the providers to adapt in their turn their offer to an educational environment. On the side of infrastructure management, the adoption of cloud resources together with a BYOD approach has a relevant organizational impact, since it almost entirely relieves the educational institution from the burden of managing computing devices, concentrating instead on the aspects that allow the students and the teachers to use their own devices in the institution premises. The accent is therefore on the network infrastructure, which, for instance, should allocate bandwidth on a per classroom basis, possibly taking into account the network load expected from a given lecture: just like today we ask for a classroom with an electronic blackboard, tomorrow we should be able to ask for a 10 Megabytes/second per student. Given that knowledge acquisition applications are indicative of such varying resource demands, based on participation, type of course etc., the need for expressing in a rich, semantically enabled manner the needed virtual service network is of high importance.

4. Development environments and abstractions

4.1. Programming abstractions

As portrayed in Sections 2.4 and 2.5, one key requirement in order to enable wider Cloud adoption and usage is to raise the level of abstraction in the creation, combination and usage of services. In order to address the aforementioned technological challenges, implemented API standards or implementation abstraction layers are necessary. Flow based programming (e.g. Nodered[3]) may also aid in this direction, especially if application templates are created that combine existing knowledge of service combination guidelines or reference architectures[4]. *Programming abstractions based on goals* that need to be achieved may also need to be defined, that are specialized in each case by relevant drivers, to adapt to specific API differentiations per provider.

4.2. Software Workbench Environments

Interactive and flexible software workbenches, that use discovery tools at the networking level and QoS requirements from the application level, can provide the tools necessary to control the lifecycle for rapid development, deployment, management and dynamic reconfiguration of complex distributed time critical cloud applications. For example, the SWITCH Interactive Development Environment (SIDE) approach provides interfaces for all user- and programmer-facing tools, by exposing a collection of graphical interfaces and APIs that tie SWITCH's services to a Web-based environment. SIDE will be engineered using fully responsive HTML5 on the front end, providing

interactivity and end-user device portability, and the back-end Web services (hooks) and APIs will be constructed using Python and tools such as Django, Flask, or similar

4.3. Cloud based IIDEs

As seen in Section 2.4, software development on the Cloud is one key future application scenario. In order to build a successful cloud-based IDE the advantages and functionality of traditional, desktop-based IDEs need to be maintained and augmented with additional features and strengths. Powerful code editors with a rich set of functionalities (e.g. highlighting, autofill, etc) should be incorporated in web browsers. Compilation and testing execution should be performed on the cloud infrastructure and, if needed, cloud providers should support deployment. Clear advantages of cloud-based IDEs include: a) the access to a very wide pool of programming tools that are maintained by the provider, thus relieving the developer from the burden to setup, configure and upgrade their programming environments, b) the ability to develop software without the use of powerful local computers, since the frequently compute-intensive tasks of the compilation and testing are performed remotely in the cloud, and c) the straightforward way to reuse code developed by other software engineers that share the same cloud environment. On the other hand, network latency can be a limiting factor, especially for small jobs while security and privacy concerns are certainly valid for valuable intellectual property as is the case of software code.

Existing platforms have made the initial steps showing potential to provide concrete implementation and execution environments for cloud-based applications. However, a number of important challenges need to be addressed for this approach to be viable. Although current solutions for cloud-based IDEs provide important new features and capabilities, they employ a subset of the features present in industrial-level desktop-based environments. In this way they are able to address a large number of development needs, focusing on specific languages and application demands, but the integration of the full set of capabilities and flexibility existing in their desktop counterparts needs to be gradually incorporated.

5. Interoperability

Interoperability is one of the major requirements driven by the scenarios, and is identified in Sections 2.1, 2.2, 2.4 and 2.5. Based on the level, we can identify two separate cases of this feature.

5.1. Technical interconnection

With respect to the technical issues, different virtualization technologies, different interfaces, and different programming languages need to be integrated without limiting the space for new technological developments. Moreover, low data portability and application portability should be overcome. An IT service platform should allow moving data, applications, and/or virtual machine images without restrictions. Service developers should be able to work with the language of their choices. While this is a separate requirement, it is also addressed through the usage of Programming abstractions mentioned in Section 4.1.

In section 4.3, the advancements in Cloud based IDEs are discussed. For that case, the existing landscape in cloud-based software development platforms has provided sufficient solutions for transferring a large number of applications to cloud infrastructures in a productive way. However, they are heavily based on ad-hoc solutions, which in several cases closely attach the developed applications to specific development environments and hosting infrastructures. Migrating projects from one platform to another or reusing components between platforms is by no means straightforward, as in several cases a number of the incorporated components are proprietary. The situation becomes even more challenging in MDE based approaches that additionally incorporate higher level of concepts and tools like models, domain specific languages, and tools for automated model management (transformation, validation, comparison, merging, refactoring etc). The *use of open standards* can provide a solid base for the development of interoperable modules, while basing the development on *open-source components* can minimize re-engineering efforts.

With relation to the joint synergies in Business Intelligence, in order to address the issues of interoperability, research is needed that combines *principles of data science, system design, business administration, and economics*.

The use of data science helps understanding the characteristics of data and the ways to manage them to generate the maximum benefits for SMEs. The use of business principles helps envisioning a way to provide an IT service analytics platform with reduced cost and benefits for all stakeholders involved. The use of system design helps integrating different solutions for the different issues of interoperability to make IT service platforms interoperable. The use of economics helps analyzing the IT service platform ecosystem with respect to the formulation of appropriate economic policies for the sustainable growth of an economy.

If these interoperability issues are resolved, a working IT service ecosystem can be envisioned, in which the stakeholders of interoperable IT platforms can interact within a sustainable economic setting. It would enable SMEs to choose among different service providers according to their needs (i.e., performance requirements, geographic spread, business goals, and costs) and to replace an existing service without any issue if needed.

5.2. Semantically Linked data

Knowledge produced, converted and produced started to be cumulated and managed on cloud. The ECLAP project on this aspect was focused in modeling and moving metadata and files on the Cloud, making them accessible as linked open data of performing art, immediately followed by Europeana Cloud, Getty, Cultura Italia, and many others. Finally, a network of linked data services is growing, and could be accessible for all. Further steps were made on modeling knowledge and providing smarter tools for general management and navigation with Linked Open Graph[22]. This is probably a starting point and not a point of arrival. The starting point of a revolution by which the knowledge will be more easily accessible and searchable, spanning across multiple RDF repositories. The main limitations now are on the *complexity by which the knowledge is moved and uploaded*, that should be improved in order to be accessible to all. *Interoperable solutions, interoperable knowledge models and tools* need to be created and followed. The RDF syntax has limitations (triples) that require multiple statements to describe knowledge that is not very simple – for example if it is of temporal duration or if there is uncertainty. The large number of triples required for realistic knowledge presentation results in performance problems. There is an urgent need for data / information / knowledge interoperation using richer metadata with formal syntax and declared semantics.

6. Legal and IPR issues

Legal and Intellectual Property Rights (IPRs) issues mainly arise with relation to cases where data aspects are involved, such as in the cases of Sections 2.1, 2.2 and 2.4. With respect to the legal issues, societies need to discuss the appropriate use of data, the changes in business processes on the society (e.g., suitability of labor contracts), and the structure of the economy. Based on the results of the discussion, legal policies need to be implemented to protect the society and guarantee sustainability of enterprises. The IT service platforms need to be able to adapt to these laws as well as to business conditions valid set by service providers. The laws and business conditions need to be translated into requirements, so that the IT service platform can handle service requests properly and law-abiding. While laws set by the administration of a country represent hard conditions for any IT service platform, business conditions set by the service owners could be dealt with by replacing a service with the one of another provider with more favorable business conditions.

For the case of shared data, like in the joint BI and Knowledge scenarios, *clear IPR agreements* need to be in place regarding the way data are utilized, referenced or exploited. Also for the case of cloud-based IDEs, data security is considered one of the most critical issues in cloud-based applications from a legal point of view. The majority of users and enterprises are reluctant to trust sensitive data to cloud environments, and this is the main reason for the development of private clouds. Software is by no means an exception in this rule. Software projects are realized by large investments and constitute a critical capital of software engineering companies.

Another major requirement for future cloud adoption, especially by the Public Sector, is the ability to have *auditable SLAs and the respective validation process of legal aspects*, potentially involving automated validation frameworks, taking under consideration relevant and necessary certifications, data placement or protection constraints.

7. Conclusions

The identification of extended application scenarios that bring new challenges to Cloud environments has led to the definition of five application areas, namely Collaborative BI platforms, Knowledge collection, Knowledge acquisition, Software development and Time critical/parametric applications. Based on the requirements of these cases, a set of domains in which advancements are needed have been concluded and analysed. The corresponding matching between these two factors is included in Table 1 and indicates the complexity involved with relation to the current technological availability.

Despite the fact that Cloud technologies have evolved considerably in the recent years and are probably one of the few cases of jumping successfully from theory to practical incorporation, still numerous steps are needed for them to reach their full potential. Based on the envisioned scenarios, these steps may include technical, conceptual or even legislative aspects that will however enable the respective domain to become more flexible, efficient and widely adopted.

Table 1. Mapping between Application Scenarios and Advancements.

	Collaborative BI	Knowledge Gathering	Knowledge Acquisition	Software Development	Time Critical Controllability
Cloud Service and App Integration	X		X		X
Development Environments and Abstractions	X			X	X
Interoperability	X	X		X	X
Legal and IPR	X	X		X	

Acknowledgements

We would like to acknowledge and thank additional contributors for the inputs, namely (alphabetic order): Dimitrios Athanasiadis, George Fylaktopoulos, George Goumas, Andrew Jones, Cees de Laat, Paul Martin, Michael Skolarikis, Vlado Stankovski, Aris Sotiropoulos, George Suciu, Ari Taal, Ian Taylor, Alexandre Ulisses, Ignacio Garcia Vega, Junchao Wang.

References

1. *Barroso, Luiz André, Jimmy Clidaras, and Urs Hölzle. "The datacenter as a computer: An introduction to the design of warehouse-scale machines." Synthesis lectures on computer architecture 8.3 (2013): 1-154.*
2. *Armbrust, Michael, et al. "A view of cloud computing." Communications of the ACM 53.4 (2010): 50-58.*
3. *Nodered Flow programming tool: http://nodered.org/*
4. *Amazon reference architectures, available at: http://aws.amazon.com/architecture/*
5. *G. Kousiouris, "Key Completion Indicators: minimizing the effect of DoS attacks on elastic Cloud-based applications based on application-level markov chain checkpoints ", to appear in Proceedings of the 4th International Conference On Cloud Computing and Services Science (CLOSER 2014), 3-5 April 2014, Barcelona, Spain*
6. *Naresh Kumar, M.; Sujatha, P.; Kalva, V.; Nagori, R.; Katukojwala, A.K.; Kumar, M., "Mitigating Economic Denial of Sustainability (EDoS) in Cloud Computing Using In-cloud Scrubber Service," Computational Intelligence and Communication Networks (CICN), 2012 Fourth International Conference on , vol., no., pp.535,539, 3-5 Nov. 2012*
7. *George Kousiouris, Andreas Menychtas, Dimosthenis Kyriazis, Kleopatra Konstanteli, Spyridon Gogouvitis, Gregory Katsaros, Theodora Varvarigou, "Parametric Design and Performance Analysis of a Decoupled Service-Oriented Prediction Framework based on Embedded Numerical Software", IEEE Transactions on Services Computing, 09 Aug. 2012. IEEE computer Society Digital Library. IEEE Computer Society*
8. *Meyer, D., (2013) The Software-Defined-Networking Research Group, IEEE Internet Computing, vol. 17, no. 6, pp. 84-87, Nov.-Dec*
9. *Muler, C., Oriol, C., Franch, X., Marco, J., Resinas, M., Ruiz-Cort, A., Rodruez, A., (2013) Comprehensive Explanation of SLA Violations at Runtime, IEEE Transactions on Services Computing, vol. 99, no. PrePrints, p. 1,*
10. *Zhao, Z., Dumitru, C., Grosso, P., de Laat, C., (2012) Network Resource Control for Data Intensive Applications in Heterogeneous Infrastructures, 2013 IEEE International Symposium on Parallel & Distributed Processing, Workshops and Phd Forum, pp. 2069-2076*

11. *Esfahani, N., Elkhodary, A., Malek, S., (2013) A Learning based framework for engineering feature oriented self adaptive software systems, IEEE transactions on software engineering, vol 39, no 11*
12. *Yan, G., Han, Y., Li, X., (2011) ReviveNet: A Self-Adaptive Architecture for Improving Lifetime Reliability via Localized Timing Adaptation, IEEE Transactions on Computers, vol. 60, no. 9, pp. 1219-1232, September*
13. *Cheng, B., Lemos, R., Giese, H., Inverardi, P., et al. (2009). Software Engineering for Self-Adaptive Systems: A Research Roadmap. In Software Engineering for Self-Adaptive Systems, Lecture Notes In Computer Science, Vol. 5525. Springer-Verlag, Berlin, Heidelberg 1-26.*
14. *Kats, Lennart CL, et al. "Software development environments on the web: a research agenda." Proceedings of the ACM international symposium on New ideas, new paradigms, and reflections on programming and software. ACM, 2012.*
15. *Tommaso Cucinotta, Fabio Checconi, George Kousiouris,Kleopatra Konstanteli, Spyridon V. Gogouvitis, Dimosthenis Kyriazis, Theodora A. Varvarigou, Alessandro Mazzetti, Zlatko Zlatev, Juri Papay, Michael Boniface, Sören Berger, Dominik Lamp, Thomas Voith, Manuel Stein: Virtualised e-Learning on the IRMOS real-time Cloud. Service Oriented Computing and Applications 6(2): 151-166 (2012)*
16. *Beydeda, Sami, and Matthias Book. Model-driven software development. Vol. 15. Heidelberg: Springer, 2005.*
17. *Stahl, Thomas, Markus Voelter, and Krzysztof Czarnecki. Model-driven software development: technology, engineering, management. John Wiley & Sons, 2006.*
18. *Kleppe, Anneke G., et al. "The model driven architecture: practice and promise." (2003).*
19. *Andreas Menychtas, Christina Santzaridou,George Kousiouris, Theodora Varvarigou, Leire Orue-Echevarria, Juncal Alonso, Jesus Gorronogoitia, Hugo Bruneliθre, Oliver Strauss, Tatiana Senkova, Bram Pellens, Peter Stuer "ARTIST Methodology and Framework: A novel approach for the migration of legacy software on the Cloud ", in Proceedings of 2nd Workshop on Management of resources and services In Cloud And Sky computing (MICAS 2013)*
20. *P. Bellini, P. Nesi, "Modeling Performing Arts Metadata and Relationships in Content Service for Institutions", International Multimedia Systems Journal, Springer, presently online at http://link.springer.com/article/10.1007/s00530-014-0366-0 , 2014, DOI 10.1007/s00530-014-0366-0 .*
21. *P. Bellini, P. Nesi, M. Serena, "MyStoryPlayer: experiencing multiple audiovisual content for education and training", International Journal Multimedia Tools and Applications, Springer, 24 April 2014,*
22. *P. Bellini, P. Nesi, A. Venturi, "Linked Open Graph: browsing multiple SPARQL entry points to build your own LOD views", International Journal of Visual Language and Computing, Elsevier, 2014,*
23. *Bartels, A., 2014. The Public Cloud Market Is Now In Hypergrowth. Sizing The Public Cloud Market, 2014 to 2020, s.l.: Forrester Research.*
24. *Eve, M., Catalano, D., Machulak, M. & Hardjono, T., 2015. User-Managed Access (UMA) Profile of OAuth 2.0, s.l.: IETF.*
25. *Hardt, D., 2012. The OAuth 2.0 authorization framework, s.l.: s.n.*
26. *Pezzini, M. & Lheureux, B. J., 2011. Integration platform as a service: moving integration to the cloud, s.l.: Gartner.*
27. *Rissanen, E., 2013. Extensible access control markup language (xacml) version 3.0, s.l.: OASIS.*
28. *NISRT Digital Repositories, available at: http://www.epset.gr/en/Digital-Content*