Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 68 (2015) 136 – 150

Procedia Computer Science

HOLACONF - Cloud Forward: From Distributed to Complete Computing

A Cloud Service Broker with Legal-Rule Compliance Checking and Quality Assurance Capabilities

Emiliano Casalicchio[a]*, Monica Palmirani[b]

[a]*University of Rome Tor Vergata, DICII, Rome, Italy*
[b]*University of Bologna, CIRFID, Bologna, Italy*

Abstract

The ICT industry, and specifically critical sectors such as healthcare, transportation, energy and government require as mandatory the compliance of the ICT systems and services with legislation and regulation, as well as with standards. In the era of cloud computing, and particularly in a public cloud scenario, this compliance management issue is exacerbated by the distributed nature of the system and by the limited control of the customer on the infrastructure/services. Also if the cloud industry is aware of this legislation/regulation compliance issue (e.g. the compliance program of Amazon, Google and Microsoft Azure), right now, there are nor reference architectures neither mechanisms capable to check and to assure, off-line and at run-time, that the compliance is guaranteed during the whole life cycle of a cloud service.

Cloud service brokerage can play an important role in law/regulation compliance management of cloud services. In this paper we propose a broker-based solution for the management of law/regulation compliance. In the specific first we define a reference architecture for a legislation-aware cloud service broker, and second we propose an autonomic manager that integrate the MAPE-K control loop with the LegEx framework for the management of the legal compliance checking lifecycle.

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

Keywords: Cloud Computing; Autonomic Computing; Legislation compliance checking; optimisation; Quality of Service;

* Corresponding author. Tel.: +39 3479374437.
E-mail address: emiliano.casalicchio@uniroma2.it

Peer-review under responsibility of Institute of Communication and Computer Systems.
doi:10.1016/j.procs.2015.09.230

1. Introduction

In the ICT industry service/systems providers, developers and integrators as well as customers should be aware that law and regulation introduce functional and non functional constraints that must be included by-design inside of the information systems and maintained during operation.

In the era of cloud computing, and specifically in a public cloud scenario, this compliance management issue is exacerbated because the customer essentially outsource data processing and storage to service providers that can be under legislation/regulation not compliant with the one of the customer[1]. Customers usually have no control on this process. Moreover, service contracts are generic and no bilateral personalisation is allowed, the services and resources could be composed and provisioned at run time and adaptation actions are continuously needed to manage workload fluctuations and component failures. Therefore are required methodologies and tools for on-line compliance checking with law/regulation and for continuous system adaptation aimed at maintaining the conformity to.

Compliance checking is triggered by many events, for example: *i)* service updated to a newest release; *ii)* change of service terms and conditions; *iii)* law/regulation changes and/or updates; *iv)* change of service composition and/or resource allocation due to utility maximization; *v)* observation of the violation of some service level thresholds; *vi)* leaving/joining of CSPs from/to the service marketplace. Concerning changes in the law, them must be propagated at run-time and this will impact: the service composition, the SLAs, the business processes. In the same way, changes in system configuration (due to adaptation actions) must be checked against law/regulation compliance and corrective actions should be taken in case of infringement. However, right now, there are no mechanism and architectures capable to check and assure the compliance during the whole life cycle of a cloud service, from on-boarding phase to initial deployment, operation and maintenance[2]. The architecture we propose is the first trying to solve this problem.

An important role in law/regulation compliance management of cloud services can be played by a cloud broker[3,4] that can work as an intermediary in the service procurement process and as a third party controller during the whole service life cycle. The broker provide services to both customers and Cloud Service Providers (CSPs), for example:

- Discovery of law/regulation compliant services;
- Checking of compliance during the service on-boarding phase and, at run time, during service evolution phase;
- Aggregation, composition, orchestration of cloud services compliant with legislation;
- Management of SLA negotiation;
- Monitoring at run time of SLA and legal rule fulfillment and actuation of adaptation actions to maintain compliance.

Let us consider, for example, the problem of procurement of governmental cloud services, that have functional and non-functional requirements imposed by European, National and local legislation and regulation. In this scenario (see Fig.1) a government agency willing to use a public cloud service could benefit from an intermediary (a broker) that offers the services mentioned above. An example of such a broker is demanded by the EU initiative Cloud For Europe[35] that funded the design and implementation of advanced cloud services to boost the uptake of cloud computing in Europe and EU public administration. Among these services there is a certified cloud service broker[36]. However, the principles behind this broker are independent from the specific sector of application and, furthermore, a cloud service broker can provide functionalities that are sector agnostic.

In literature there are many research work on cloud service brokerage, addressing different issues: interoperability[5,6], service discovery and matching[7], quality assurance and optimization[6,8,9,10] and legislation compliance[1]. Despite these efforts, at the best of our knowledge this is the first paper that proposes a reference architecture for a cloud broker with self adaptation capabilities to comply with law/regulation changes.

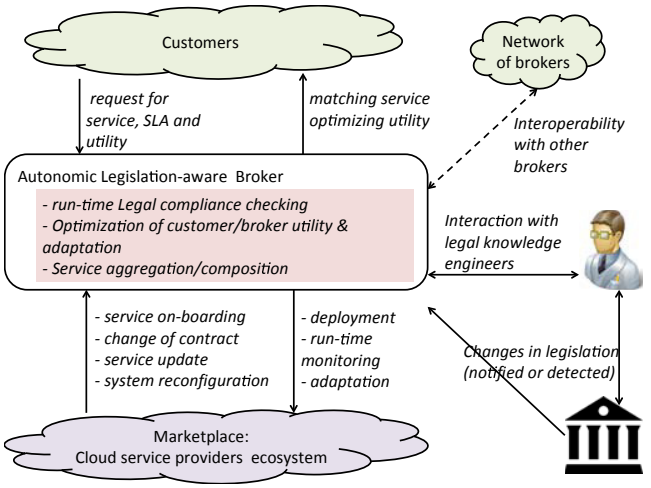

Fig. 1. The reference scenario.

Our work contributes to the literature as follow:

- We define a reference architecture for a self-adaptable and legislation-aware cloud service broker
- We propose an autonomic manager with legal compliance checking capabilities. The autonomic manager integrates the MAPE-K control loop with the Legislation Execution framework for the management of the legal rule compliance-checking life cycle.

These preliminary results are produced in the context of the industrial research framework mentioned in [35].

The paper is organized as in what follow. Related works on cloud service brokerage are discussed in Section 2. Section 3 presents the LegEx framework for compliance checking. The broker reference architecture is introduced in Section 4 and in Section 5 is presented the legislation-aware autonomic controller we proposed. Section 6 concludes the papers.

2. Related works

According to the NIST definition[3] a Cloud Broker is "*an entity that manages the use, performance, and delivery of cloud services, and negotiates relationships between Cloud Providers and Cloud Consumers".* NIST identifies three categories of cloud brokerage services: Intermediation, Aggregation and Arbitrage. Intermediation refers to the ability to enhance a given service by improving some specific capability, and providing value-added services to cloud consumers. Aggregation refers to the capability to combines and integrates multiple services into one or more new services. And arbitrage is similar to service aggregation except that the services being aggregated are not fixed. A similar definition is given by Gartner[11] while the concept of autonomicity is stressed in[4] that defines a cloud service broker as *"an entity responsible for: automatic resource provisioning and management across multiple clouds; automatic deployment of application components; and scheduling and load balancing of the incoming requests to the allocated resources".*

The problem of service brokering has been addressed from different perspectives in the literature but, at the best of our knowledge, no research work address explicitly the problem of legal compliance checking, except in[1] where the authors propose a distributed cloud proxy for monitoring and controlling the cloud service consumption. The aim of the proxy is to enable compliance to all privacy, legal, and regulatory issues regarding the service consumption. The proxy is comparable to an application layer gateway for cloud computing service and can not be exactly classified as a broker.

The need of brokering mechanism particularly arises in Cloud Federation architectures, such as Intercloud[4], the first approach going in the direction of building a unified platform composed by federated providers that can exchange information through super-entities. Service brokering is expected also to facilitate cloud adoption simplifying the matching of users need[7], to rise trust in cloud computing and to facilitate the procurement of cloud services in the public sector providing added-value services[6,12] .

In[33] is presented an architecture for a federated Cloud computing environment named InterCloud to support the scaling of applications across multiple Cloud providers. The proposed solution use a Cloud Broker for mediating between service consumers and Cloud coordinators for an allocation of resources that meets QoS needs of users.

In[13] is described the concepts of cloud bursting and cloud brokerage and is discussed the open management and security issues associated with the two models. It also presents a possible architectural framework capable of powering the brokerage based cloud services that is currently being developed in the scope of OPTIMIS, a EU FP7 project. Another result of the OPTIMIS project[34] is a schema definition and usage mechanism (CPDS) that includes various levels of legal information that is necessary for automating the process of Cloud provider selection and data outsourcing. Thus the legal constraints may be checked in an automated and machine understandable.

Cloud Agency implements a multi-agent brokering mechanism[14] that is vendor agnostic and allows for the deployment of mOSAIC applications on any Cloud infrastructure. In[5] the authors present the architecture of the Broker Agent and its implementation in Cloud Agency for provisioning of brokering service.

In[8] the authors propose a Cloud brokering approach that optimises placement of virtual infrastructures across multiple Cloud providers (each one with a different infrastructure offer and pricing policy) and also abstracts the deployment and management of infrastructure components in these Clouds.

STRATOS[10] is another cloud broker service that permits to deploy and manage cloud applications on multiple providers, based on requirements specified in higher level objectives. STRATOS solve a multi-objective optimisation problem and address the runtime adaptation issue.

In[15] the authors consider the service brokering at IaaS level as a mean to realizing delegation in cloud federations, that is to allow IaaS providers leveraging the capabilities available in a federation.

QBROKAGE[12] address the problem of scalability and vendor lock-in exploiting only info public available from service providers. The proposed solution allow the deployment of applications on VMs running on multi-clouds. The authors propose a genetic service selection algorithm.

In[6] the authors propose an approach similar to the one presented in this paper. Their broker receives from the customer a call for proposal indicating functional and not functional requirements (SLA) and return the as result the best proposal, i.e. the best offer from providers. The SLA consider price, time unit, a rating indicating the best-accredited provider, the minimum accepted availability, but does not consider explicitly legal rules in the adaptation process. Moreover is not addressed the run time adaptation problem.

In[7] the authors addressed the problem of cloud service matching proposing and OWL-S based cloud services broker. The complex constraints considered are on service location, bandwidth, storage, cost and usage. This solution can be used also to solve the problem interoperability due to a non standard way for exposing providers capabilities. Also if their approach uses semantic reasoning, as the one we propose, the authors do not address the legal compliance problem and the run time adaptation problem.

3. The Legislation Execution framework

Cloud computing gives rise to several legal barriers expressed in numerous European and national regulations, e.g. data protection policy, rules on data archiving, eId regulations, transfer of personal and sensitive data, data retention, contractual clauses (e.g. penalties and reparations, contractual liability, jurisdiction, etc.), consumer law.

However, in any domain, the law not only includes obligations but also it constitutes new business processes and so, also, in the cloud computing scenario, the law has a strong impact on the business processes.

The variety of legislations (European, national, regional, specific domain) makes it necessary to implement a framework for managing legal compliance checking and thus making it possible to achieve two main goals:

- to detect ex-ante the measures needed to prevent a cloud service from infringing laws, regulations, policies, contracts, or service legal agreements (SLAs);

- to alert cloud service providers and cloud service broker in case the legislation (or in general legal documents) should introduce new service requirements, new privacy or security policy specifications or new business processes.

For producing a feasible and lawful legal compliance checking it is necessary a defeasible, deontic and temporal logic model connected with the legal original texts in order to produce reports able to justify the outcomes. Auditable, scalable, computable, effective, simple, interactive, interoperable, traceable, standard legal compliance checking architecture is possible using a set of tools capable to model the norms in formal manner.

These needs are addressed by a *Legislation Execution* (LegEx) framework that aims at:

- Managing the legal resources lifecycle in the evolving legislative scenario;
- Detecting the legal barriers to cloud computing services; suggesting optional solutions to experts for minimizing the legal risks in order to define proper policies (e.g., privacy policies, identity policies, security policies);
- Enabling the business model component to interoperate with the workflow module that manages the cloud computing brokerage network for storing legislative changes and so, when possible, to deliver the proper service and behavior among the different nodes in the brokerage network.

The main macro-components of the LegEx framework are (see Fig. 2):

- A legal sources modelling dashboard based on XML standards (Akoma Ntoso[16] and LegalRuleML[17], LIME[18] Web editor technology) with a temporal legal model[19];
- A legal knowledge extractor that uses NLP (natural language processing) tools and legal ontologies oriented to minimize the legal sources modeling and also to improve the Semantic Web query on legal documents;
- A legal compliance checking module based on business process interaction using legal reasoning engine (SPINdle[20]) and business process modeling (BPM editor). The approach used is both for forward compliance checking in order to forestall violations and backward compliance checking alerting the data-protection authority if a violation should occur[21,22];
- A business process component for modeling and tuning the business processes concerning the cloud services on the basis of communication with legal compliance checking in the modeling phase. In the run-time phase the BPMN 2.0 DB is an input source of rules;
- A component for the integration with the cloud computing infrastructure for run time compliance checking.

The LegEx integrated framework is able to manage four main crucial issues in legal compliance checking, that are:

- Management of Changes,
- Legal Knowledge Discovery,
- Legal Reasoning,
- Business Process Modeling.

Legal document change over time, especially as that applies to acts, regulations and contracts which by nature are variable and subject to frequent modification, significantly affecting coordination between the text and the rules that need to be remodeled. The LegEx *Management of Changes* capability allows to properly deal with such changes.

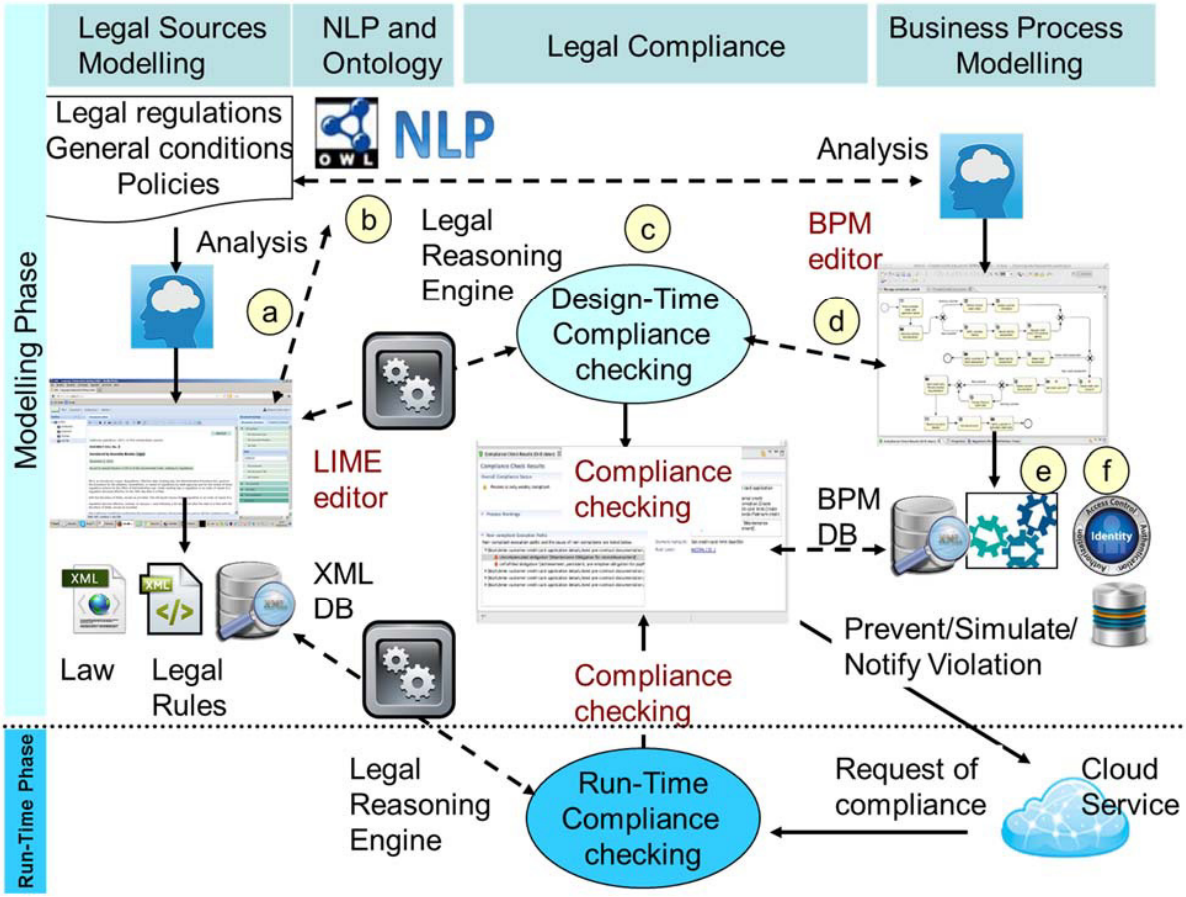

Fig. 2. Legislation Execution Framework architecture.

Legal knowledge discovery within legislative documents and in general within legal documents is managed by means of NLP tools that support the task of detecting legal knowledge contained in the text. Legal ontologies are needed in connecting abstract legal concepts in a specific domain with NLP linguistic resources. This part of the system also manages ontologies over time using ontology learning techniques. Finally the same NLP/ontologies techniques are used to provide legal knowledge engineers with a tool to detect the new norms emerging in the field, and so to enable experts to also update the business processes repository.

The *Legal Reasoning* engine (that uses the legal sources previously marked up with the web editors) is based on defeasible and temporal logic, specific to legal domain, and it also is scalable and computable with the relevant volume of rules. It guarantees legal compliance checking by means of a specific algorithm allowing to answer the queries submitted by cloud service providers or the by the service brokerage infrastructure. When a fact or a service is required, the cloud computing infrastructure asks the legal reasoning engine to verify the legality of the operation using, among the other resources, the general conditions of the contract, the relevant case-law decisions, soft-law policies. The result of legal reasoning is a report detailing violation, reparations, and possible alternative solutions that need the interpretation[23] by the decision-maker (cloud actors).

With legal reasoning is necessary to *Model Business Processes* for guaranteeing the correct application of the technical operations, events, processes connected with cloud computing services. To that end, we have a special editor for modeling business processes using BPMN 2.0. This module is invoked in each legal compliance-checking action in order to see if the lawful obligations are also compliant with the real applicative scenario. Legal reasoning

is also invoked when a law changes for checking if the business processes are consistent with the new legislative modified scenario. If so, an alert is produced for the business process designer in order to updated the workflow component that works with the system/service checked.

3.1. The legal compliance checking life cycle

The legal compliance checking life cycle is based on two main phases: the *Modeling phase*, or preparatory off-line phase; and the *Run-time phase*, or query phase.

In the *Modeling phase* as soon as a new service is introduced by law (e.g. new informed consent procedure, new archiving method, etc.), the models of the business processes that are regulating all the cloud services should be refined according to this modification. The legal knowledge resources where to discover obligations, permissions, rights, prohibitions, penalties, reparations are mainly the law, the contracts, the case-law documents.

In the *Run-time phase* the cloud service broker (and cloud service providers joining the marketplace) must invoke the Legislation Execution for checking if the specific service requested is lawful and policy compliant. When a law is modified a demon detects all the connected texts, legal rules and legal ontology concepts that need to be updated with the support of a human legal expert. Using the legal reasoning engine the framework also detects the business processes needing to be updated. This approach permits to avoid the violation of the legal provisions, from the beginning of the system operation, by adopting a legal-by-design approach.

Another relevant feature of the LegEx framework is to use both backward and forward legal compliance checking integrated with BPM. *Backward legal compliance-checking* detects the violation after that the process is activated. It is an ex-post analysis of the log file for detecting if something was not properly managed. The Cloud service providers should pass to the BPM engine all the events as log file. The *Forward legal compliance-checking* approach allows to verify the compliance checking both in the design/modeling phase and in the run-time phase. This permits in the design/modeling phase to define correct BPM processes according to the law regulations and in the run-time phase to prevent the violation of the law.

Both in backward and forward approaches the steps of the life cycle are the following:

1. A legal expert selects resources (all the legislation/regulation/policy/contract/case-law) pertinent with the domain of the application (e.g., all the privacy regulation) using a particular tools called Eurnomos;
2. A legal knowledge engineer provides the first core of legal domain ontology of the domain (e.g., privacy, digital identification, cloud computing, etc.);
3. The computational linguists and the engineers tune the NLP tools on the base of the language of the legal documents;
4. Using the web specialized editors LIME and RAWE (LIME editor for Rules[24]) the legal expert models the text transforming it into legal rules using deontic and defeasible logic. In this phase the editors are supported by NLP tools and by legal ontology.
5. The legal engine SPINdle produces the compliance checking with the existing Business processes that represent the cloud computing services. In this phase are received requests from the broker to check the compliance of aggregated and mediated services, both in the on-boarding that in the evolution phase.
6. The legal compliance checking module Regorus[25] provides a report about the business processes not in line with the law. The report produces a list for priority, of justifications and evidence, of the legal original text related to. The report includes also some possible solutions for correcting the business process that breaks the law (e.g., introduce a new consent module, a new information web page). Right now the report is validated by a legal experts in the LIME editor in order propagate the correct rule. The main challenge in this research is to automate this phase producing a report that will feed directly the broker modules in charge of adapting the services (see Sec.4).
7. Periodically the Eunomos crawler detects the law/regulation changes. New law or fragment of law/regulation/legal material is detected and the NLP tools extract from the new law the modifications that are applied to the existing legal documents database for producing the updated version. A new updated legal ontology is produced using learning ontology techniques and the rules affected by the modifications are easily retrieved by the legal rule base and the legal knowledge expert tunes them using RAWE editor.

Steps 1-4 characterize the preparatory Modeling phase, while steps 5-7 describe the run-time query phase and are cyclically repeated.

4. The Reference Architecture

The solution we propose can be classified, according to the NIST definition, as an Intermediation-Aggregation broker and support four main capabilities[26,27,28]:

- *Discovery* of services that are compliant with functional and non-functional requirements imposed by law and regulation.
- *Aggregation* of cloud services in order to meet customer requirements. The broker can provide one or more added-value {\em aggregated services} implementing complex functionalities or can provide {\em mediated services}, that is single functionality.
- *Quality Assurance* of services by means of off-line and run-time verification.
- *Optimization* of broker and/or customers utility. Optimization is related to the maximization of a utility function subject to quality of service and legislation/regulation constraints.

The high level architecture we proposed is organized around three main modules as shown in Fig. 3. Two of them, the **Service Provider Interfaces** (SPI) and the **Service Consumer Interfaces** (SCI) modules, implement the access point for the service providers and the service consumer. The SPI includes: an *SPs Account Management* component that provides the interface for SP registration and authentication and it manages SP registration data; and an *SP Service Registration* component that is responsible for the enrolment of new services in the broker, i.e., it allows to start and to manage the service on-boarding phase and the service evolution during the service life cycle. This later component interacts with the *Quality Assurance & Optimization* module. The SCI includes a *Service Discovery and Presentation* component provides: a graphical interface allowing SCs to specify their functional and non-functional requirements; an automated service discovery tool to search in the local and remote service metadata registries a cloud service matching the SC needs. This tool returns to the users the (possibly aggregated) services matching their requirements, sorted by some utility criterion. The services are ranked on the basis of an acceptance score. Moreover, the SCI includes an *SC Contract Management* component responsible for storing and maintaining information about the contracts stipulated between service consumers and providers through the broker.

These end users functionalities are delivered as Software-as-a-Service applications and are not further discussed in the paper because not relevant.

The third module, the *Quality Assurance & Optimization* (QAO), implements the autonomic features of the broker for quality assurance and optimization. This module is in charge of:

i) to guarantee the on-boarding of compliant services,
ii) to maintain the desired level of QoS and legislation compliance at run-time, and
iii) to optimize the utility of service providers and/or consumers and/or the broker.

The QAO functionalities are implemented by several software components that are described in what follow.

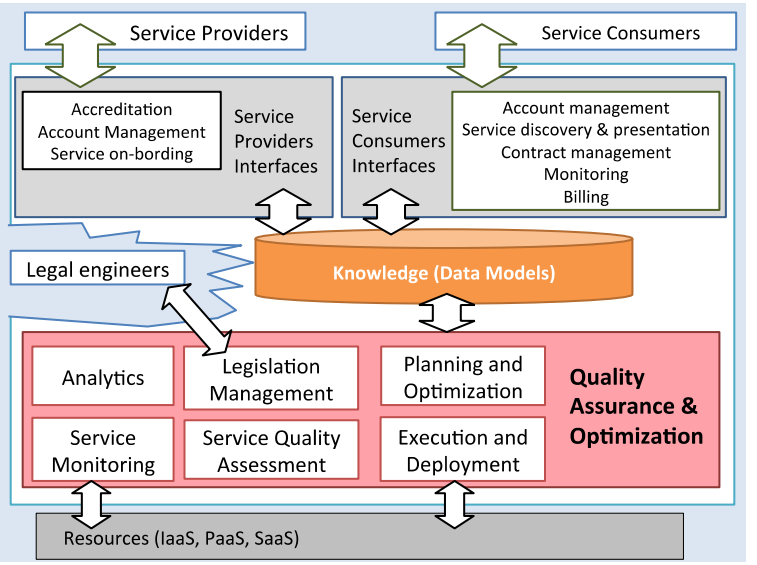

Fig. 3. The high level architecture of the legislation-aware autonomic cloud service broker. The figure shows the functional modules of the broker.

The **Service Quality Assessment** (SQA) component provides the capabilities for off-line verification of service compliance with the constraints imposed by the law and regulation. This module takes as input all the information provided by the service provider during the service registration process.

The SQA component use the functionalities offered by the **Service Legislation Management** (SLM) component that interact with the LegEx framework (described in detail in Section 3) to automatically check the compliance to legislation in term of non-functional requirements, business processes, standard adherence and other constraints. The SLM module is also in charge to constantly monitor and analyze the law and regulation landscape, which is the phase 7 of the legal compliance checking life cycle.

The **Service Monitoring** component is responsible for the continuous monitoring of the cloud services, which could be aggregated or mediated. Monitoring is related to the SLAs metrics described in the broker knowledge base and is mainly implemented using monitoring APIs provided by SPs and/or dedicated monitoring agents specifically developed and deployed.

The goal of the **Analytics** component is to put in place data analysis techniques to detect or better predict SLAs violations. The component inspects the monitored data, computes direct and indirect metrics, and determines if the SLAs are violated, and/or forecasts the short term value for that metrics to predict and therefore enabling system adaptation to avoid SLA violation.

The **Planning and Optimization** component is notified by the Analysis component and the SQA component in case of SLAs/law/regulation violations. This component is then in charge of to determine plan an adaptation policies that could involve: service re-configuration, resource provisioning, traffic re-routing. The service adaptation policy has, of course, the goal of maintaining the compliance with law and regulation, but also to guarantee that all the non functional constraints are satisfied and the broker and/or customer utility is maximized. Our solution uses a linear programming approach to optimize the service configuration[29].

The **Execution & Deployment** component is in charge to implement the adaptation policies needed to meet customers requirements; to interact with IaaS, PaaS or SaaS providers to notify SLAs/SLOs violations; to instantiate/deploy resources to run the aggregated services implemented by the broker. In this context it doesn't matter if the broker has its own IaaS infrastructure or is hosted on a third party infrastructure.

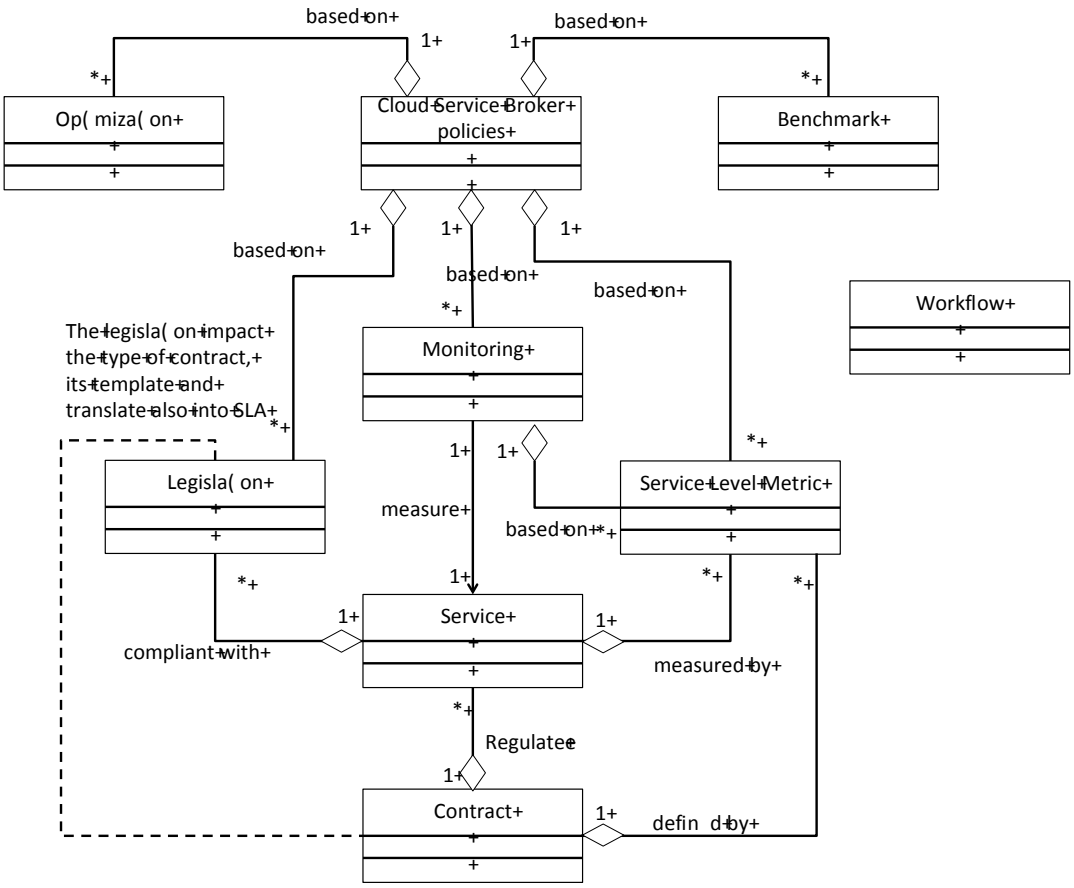

Fig. 4. The data model of the legislation-aware autonomic service broker

All the broker components and the functionalities they implement rely on a shared knowledge base defined by complex and interdependent data models. In the following, we provide a high-level description of the data models and their relationships (see Fig.4). The *Legislation* (LEG) data model describes the legislation for the EU member state running the broker. LEG defines and implements the Agile Data Model for Legislative Domain using Akoma Ntoso[16] and LegalRuleML[17], which are both candidates as OASIS international standards. The *Service Level Metrics* (SLM) data model describes the metrics that should be monitored. Metrics of interest can be either simple or aggregated. For each metric a description of the necessary information to implement and activate the related monitoring will be provided. The *Service* data model describes the cloud services accredited with the broker on the basis of the Service Configuration template. The template also contains the description of non-functional requirements linking the service level metrics and legislation data models. The *Contract* data model describes contracts on the basis of a specific template. A contract is effective for one or more cloud services and specifies a service level agreement (SLA). Therefore, the Contract data model links the Service Description, the Service Level Metrics, and the Legislation data models. The *Monitoring* data model describes the data monitored for each contract and for the related services. The Monitoring data model links contract and service description data models (i.e., links metrics and legislation to be monitored). The *Optimization* data model describes the optimization rules and utility functions. The *Benchmark* data model describes tests, certification schemes, and benchmarks needed for service on-boarding and to verify the service migration success. The data model will be also capable to describe the results of the tests for possible future use (e.g., statistical analysis, comparison, SLA improvement). The *Cloud Service Broker*

Policies data model describes the rules that should be satisfied for SPs and cloud services certification and the actions that should be taken in case of violation of the certified requirements. These policies are based on benchmark data, service level metrics, legislation data, monitoring data, optimisation data and service descriptions. The *Workflow* data model describes the workflows supporting the brokering processes (e.g., legislation change propagation, seamless service migration, on-boarding, certification) and the lawful business processes.

5. The Legislation-aware Autonomic Manager

The quality assurance and optimization module of the broker is essentially the controller of the whole architecture. The mapping of the broker components on the classical MAPE-K manager is illustrated in Figure 5. The Analyzer and the Monitor integrate the legislation execution/management framework and therefore are enhanced with legal rule-aware capabilities. This integration introduces, in all the phases of the autonomic cycle, new activities and challenges that are discussed in what follow and summarized in Table 1.

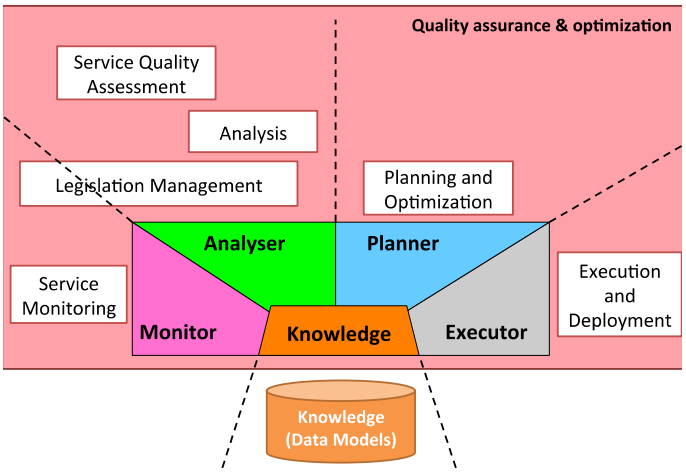

Fig. 5. The Legislation-aware Autonomic Manager

Table 1. Legislation-aware Autonomic Manager activities.

Phases	Activities
Monitoring	Monitoring of Service Level metrics
	Monitoring of legislation compliance
	Monitoring of Cloud service provider changes / Marketplace changes
Analysis	Analysis of QoS metrics, detection/forecasting of SLO violations
	Analysis of changes in legislation to assess compliance
Planning	SLA violation driven reconfiguration
	SLO change driven reconfiguration
	Cloud SP / Marketplace driven reconfiguration
	Business process change driven reconfiguration

5.1. Monitoring

Compliance checking is triggered by many events, mainly generated by service providers and government. To cope with these different events, a Legislation-aware Manager should run three different but correlated monitoring activities (see Table 2):

- *Monitoring of service level metrics*, or QoS metrics, is related to the collection of data for the assessment of the performance of the external services (the services the broker integrate or simply mediate) and for self assessment. In the first case, the monitoring of third party cloud services require the use of third party cloud services monitoring API and/or the deployment of monitoring agents. The availability of monitoring API is a requirement for cloud service provider willing to join the marketplace. Inter-cloud and cross layer monitoring are other two challenges already recognized in literature[30]. For what concern the self-evaluation, the broker has to monitors:
  - *i)* the performance of added-value services functional to the usage of the broker by means of customers and providers (e.g registration, on-boarding, certification, monitoring, discovery);
  - *ii)* the performance of the mediated cloud services the broker orchestrate with the goal of maximizing broker/client utility (e.g. the broker distributes the requests from a customer among two or more cloud providers to balance cost and availability);
  - *iii)* the performance of the aggregated services the broker orchestrate.
- *Monitoring of legislation changes*. Right now, in the legislation execution framework, if a new law/regulation is produced or an existing one is updated, the text of the law/regulation is properly coded and modeled by legal engineers and then the compliance of a system is evaluated against the legal rules (see Legal source modelling and NPL and Ontology phases in Fig. 2). This approach should be split in two phases. Detection of a change and analysis of the change. The detection of the change is carried on in the monitoring phase. The legislation change monitoring module should only detect if a new legal document is produced and should acquire the document in a well formed format. And additional step should be the classification of the document as a new law/rule or as an update (modification) of an existing one. The remaining of the work is delegated to the analyzer.
- *Monitoring of cloud service changes*. The monitor should also detect if a cloud service provider update its services or the term and condition of the contract. Three approaches can be used for the monitoring of legislation changes and cloud service changes: a proactive approach, which is the broker run agents to discover the changes; a reactive approach, that is the broker is notified by providers or legal/regulation body of the change; or a mixed approach.

Table 2. Events triggering compliance checking, root cause actors and monitoring activities. This table define the relation among who, and what event, activate a compliance checking action, and the related monitoring activity to catch the event.

Root Cause Actor	Event	Monitoring Activity
Government	Law/regulation changes and/or updates	Monitoring of legislation changes
Service provider	Service updated to a newest release	Monitoring of cloud service changes
Service provider	Change of service terms and conditions	Monitoring of cloud service changes
Service provider	Change of service composition and/or resource allocation due to utility maximization	Monitoring of service level metrics
Service provider	Violation of service level thresholds (SLO)	Monitoring of service level metrics
Service provider	Leaving/joining of CSPs from/to the service marketplace	Monitoring of cloud service changes

5.2. Analysis

The Analyzer component is in charge of analyzing both QoS metrics and legislation/regulation changes:

- *Analysis of QoS metrics* to compute QoS metrics and to evaluate SLAs violations. We assume a SLA includes a set of SLOs that map QoS constraints on QoS metrics. SLOs violation can be detected (with a passive approach)

or forecasted and therefore avoided/mitigated (with an active approach). This is the standard role of the analyser and many solutions has been discussed and proposed in the literature[31].

- *Analysis of changes in legislation*. This is the compliance checking phase operated on the basis of: the legal documents collected by the monitor; the information collected on update of a services; the information collected on the update of terms & conditions (i.e. the service contract). The compliance checking is evaluated against the business processes operated directly by the broker (that is the added values services) and against the services mediated by the broker. Therefore, to integrate *Phase 6* of the LegEx life cycle into the autonomic manager, the LegEx should produce three different outputs:
  - *i)* a new set SLO_*^r of SLOs imposed by the new law/regulation r. This set is empty if the changes in legislation do not impact the QoS level. * stands for *low* or *up* in case the SLO is an upper bound or a lower bound on the metric, respectively.
  - *ii)* A new business processes $\mathcal{C}^r$ imposed by the new law/regulation. This process is produced only if the existing one is incompatible with the new law/regulation.
  - *iii)* A set of service providers P_{exit}^r that are no more compliant and must be excluded from the selection, and a set of new entry service providers P_{new}^r. These two lists can be empty if changes does not impact the compliance and/or if no new providers join the market place. The set $P^r = P_{new}^r \cup \{ P^r \setminus P_{exit}^r \}$ of service providers that are compliant with the current law and regulations will be elaborated by the planner.

5.3. Planning

The Planner component of the MAPE-K cycle is in charge to evaluate a new system configuration to reestablish legal compliance and QoS assurance. The sets mentioned above, SLO_*^r, $\mathcal{C}^r$, P_{exit}^r and P_{new}^r are the inputs for the planning phase. If all the sets are empty no adaptation is needed. If only the set of new providers is non empty the reconfiguration can be postponed when will be triggered an adaptation for detected or foreseen violations. Otherwise, a new system reconfiguration must be evaluated.

There are four cases for reconfiguration actions:

- If SLOs are violated for mediated services a new selection must be determined by the broker to satisfy customer requests. The broker must notify, through the Executor, the service providers that violated the SLOs and should apply a policy to determine if the faulty cloud service should be removed from the list of certified services. The service providers will take the necessary corrective actions.
- If a new set of SLOs SLO_*^r imposed by the new law/regulation is produced an adaptation of service configuration should be operated to assure the new constraints are satisfied.
- If the law/regulation impose a new business process structure $\mathcal{C}^r$ the aggregated service will be adapted accordingly.
- If the set P^r of service providers compliant with the current law and regulations change, both mediated and aggregated services will be reconfigured.

The reconfiguration is also oriented to optimize the utility of the broker and of the customers. In our previous work[29] we propose a QoS-aware flow-based run-time adaptation model for service oriented system. The adaptation model is based on linear optimization problem that consider constraints introduce by laws and regulations. In the specific constraints on:

- the service level agreement SLO_*^r;
- the implementation of the service that can be selected P^r ; and
- the structure of the composed service $\mathcal{C}^r$.

6. Concluding remarks

In this paper we presented the preliminary results produced in the context of the industrial research framework mentioned in http://www.agid.gov.it/cloudforeurope. In the specific we propose an integrated approach for the solution of the autonomic management of run-time legal-rule compliance of cloud services. First, we define a reference architecture for a self-adaptable and legislation-aware cloud service broker. We define all the broker components and we describe in detail the functionalities of the Quality Assurance and Optimisation module.

Therefore, we propose an autonomic manager with legal compliance checking capabilities. The legal compliance aware autonomic manager is obtained integrating the legal compliance checking life cycle with the MAPE-k control loop. Essentially, the legislation management phases (implemented by the LegEx framework) are integrated with the monitoring and analysis phases of the MAPE-K cycle.

This logical integration lead also to design the physical integration of the MAPE-K components with the LegEx framework components, as described in Section 5. The LegEx solution is a complex, integrated framework available as Web service, but it is modularized in separate off-line software components and additionally it is possible to invoke specific modules using API RESTful. This modularity and the web service technology adopted facilitate the integration in an autonomic manager such as[32].

References

1. Thatmann, D., Slawik, M., Zickau, S., Kpper, A.. Towards a federated cloud ecosystem: Enabling managed cloud service consumption. In: Vanmechelen, K., Altmann, J., Rana, O., editors. Economics of Grids, Clouds, Systems, and Services; vol. 7714 of Lecture Notes in Computer Science. Springer Berlin Heidelberg.
2. Kourtesis, D., Bratanis, K., Friesen, A., Verginadis, Y., Simons, A., Rossini, A., et al. Brokerage for quality assurance and optimisation of cloud services: An analysis of key requirements. In: Lomuscio, A., Nepal, S., Patrizi, F., Benatallah, B., Brandi, I., editors. Service-Oriented Computing ICSOC 2013 Workshops; vol. 8377 of Lecture Notes in Computer Science. Springer International Publishing.
3. Liu, F., Tong, J., Mao, J., Bohn, R., Messina, J., Badger, L., et al. NIST cloud computing reference architecture. NIST special publication 2011;500:292.
4. Grozev, N., Buyya, R.. Inter-cloud architectures and application brokering: taxonomy and survey. Software: Practice and Experience 2014; 44(3):369–390.
5. Amato, A., Venticinque, S.. Multi-objective decision support for brokering of cloud sla. In: Advanced Information Networking and Applications Workshops (WAINA), 2013 27th International Conference on. 2013, p. 1241–1246. doi:10.1109/WAINA.2013.149.
6. Amato, A., Di Martino, B., Venticinque, S.. Cloud brokering as a service. In: P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC), 2013 Eighth International Conference on. IEEE; 2013, p. 9–16.
7. Ngan, L.D., Kanagasabai, R.. Owl-s based semantic cloud service broker. In: Web Services (ICWS), 2012 IEEE 19th International Conference on. 2012, p. 560–567. doi:10.1109/ICWS.2012.103.
8. Tordsson, J., Montero, R.S., Moreno-Vozmediano, R., Llorente, I.M.. Cloud brokering mechanisms for optimized placement of virtual machines across multiple providers. Future Gener Comput Syst 2012;28(2):358–367. URL: http://dx.doi.org/10.1016/j.future. 2011.07.003. doi:10.1016/j.future.2011.07.003.
9. Houidi, I., Mechtri, M., Louati, W., Zeghlache, D.. Cloud service delivery across multiple cloud platforms. In: Services Computing (SCC), 2011 IEEE International Conference on. 2011, p. 741–742. doi:10.1109/SCC.2011.107.
10. Pawluk, P., Simmons, B., Smit, M., Litoiu, M., Mankovski, S.. Introducing stratos: A cloud broker service. In: IEEE CLOUD. 2012, p. 891–898.
11. Gartner, . Cloud services brokerage (csb). http://www.gartner.com/it-glossary/cloud- services- brokerage- csb/; 2015.
12. Anastasi, G.F., Carlini, E., Coppola, M., Dazzi, P.. Qbrokage: A genetic approach for qos cloud brokering. In: Cloud Computing (CLOUD), 2014 IEEE 7th International Conference on. IEEE; 2014, p. 304–311.
13. Nair, S., Porwal, S., Dimitrakos, T., Ferrer, A., Tordsson, J., Sharif, T., et al. Towards secure cloud bursting, brokerage and aggregation. In: Web Services (ECOWS), 2010 IEEE 8th European Conference on. 2010, p. 189–196. doi:10.1109/ECOWS.2010.33.
14. Amato, A., Di Martino, B., Venticinque, S.. Evaluation and brokering of service level agreements for negotiation of cloud infrastructures. In: Internet Technology And Secured Transactions, 2012 International Conference for. 2012, p. 144–149.
15. Villegas, D., Bobroff, N., Rodero, I., Delgado, J., Liu, Y., Devarakonda, A., et al. Cloud federation in a layered service model. J Comput Syst Sci 2012;78(5):1330–1344.
16. UNDESA,.AkomaNtoso,XML for parliamentary, legislative and judiciary documents. http://www.akomantoso.org; 2015.
17. OASIS, . LegalRuleML, enabling legal arguments to be created, evaluated, and compared using rule representation tools. https://www.oasis-open.org/committees/legalruleml/; 2015.
18. CIRSFID, Unibo,. Language independent markup editor (LIME). http://lime.cirsfid.unibo.it; 2015.
19. Palmirani, M., Governatori, G., Contissa, G.. Modelling temporal legal rules. In: The 13th International Conference on Artificial Intelligence and Law, Proceedings of the Conference, June 6-10, 2011, Pittsburgh, PA, USA. 2011, p. 131–135.

20. Lam, H.P., Governatori, G.. The making of SPINdle. In: Rule Interchange and Applications. Springer; 2009, p. 315–322.
21. Tosatto, S.C., Kelsen, P., Ma, Q., Kharbili, M.E., Governatori, G., van der Torre, L.W.N.. Algorithms for tractable compliance problems. Frontiers of Computer Science 2015;9(1):55–74.
22. El Kharbili, M.. Business process regulatory compliance management solution frameworks: A comparative evaluation. In: Proceedings of the Eighth Asia-Pacific Conference on Conceptual Modelling - Volume 130; APCCM '12. Darlinghurst, Australia, Australia: Australian Computer Society, Inc.
23. Boella, G., Janssen, M., Hulstijn, J., Humphreys, L., van der Torre, L.. Managing legal interpretation in regulatory compliance. In: Proceedings of the Fourteenth International Conference on Artificial Intelligence and Law; ICAIL '13. New York, NY, USA: ACM. ISBN 978-1-4503-2080-1; 2013, p. 23–32.
24. Palmirani, M., Cervone, L., Bujor, O., Chiappetta, M.. Rawe: An editor for rule markup of legal texts. In: RuleML (2). 2013, .
25. Governatori, G., Shek, S.. Regorous: a business process compliance checker. In: International Conference on Artificial Intelligence and Law, ICAIL '13, Rome, Italy, June 10-14, 2013. 2013, p. 245–246.
26. Simons, A.J.H., Bratanis, K., Kourtesis, D., Paraskakis, I., Veloudis, S., Verginadis, Y., et al. Advanced service brokerage capabilities as the catalyst for future cloud service ecosystems. In: Proceedings of the 2Nd International Workshop on CrossCloud Systems; CCB '14. New York, NY, USA: ACM.
27. Fowley, F., Pahl, C., Zhang, L.. A comparison framework and review of service brokerage solutions for cloud architectures. In: Lomuscio, A., Nepal, S., Patrizi, F., Benatallah, B., Brandi, I., editors. Service-Oriented Computing ICSOC 2013 Workshops; vol. 8377 of Lecture Notes in Computer Science. Springer International Publishing. ISBN 978-3-319-06858-9; 2014, p. 137–149.
28. Broker@Cloud, . Enabling continuous quality assurance and optimization in future enterprise cloud service brokers (brokeratcloud), fp7-ict eu project. http://www.broker-cloud.eu; 2012.
29. Casalicchio, E., An Autonomic legal-rule aware cloud service broker. In: 2015 IEEE International Conference on Cloud and Autonomic Computing (ICCAC). IEEE; 2015, .
30. Aceto, G., Botta, A., De Donato, W., Pescape`, A.. Cloud monitoring: A survey. Computer Networks 2013;57(9):2093–2115.
31. Davis, I., Hemmati, H., Holt, R., Godfrey, M., Neuse, D., Mankovskii, S.. Storm prediction in a cloud. In: Principles of Engineering Service-Oriented Systems (PESOS), 2013 ICSE Workshop on. 2013, p. 37–40. doi:10.1109/PESOS.2013.6635975.
32. V.Cardellini,E.Casalicchio,V.Grassi,S.Iannucci,F.LoPresti,andR.Mirandola,"Moses: A framework for qos driven run time adaptation of service-oriented systems," Software Engineering, IEEE Transactions on, vol. 38, no. 5, pp. 1138–1159, 2012.
33. R. Buyya, R. Ranjan, R.N. Calheiros, "InterCloud: Utility-oriented Federation of Cloud Computing Environments for Scaling of Application Services", Proceedings of the 10th International Conference on Algorithms and Architectures for Parallel Processing - Volume Part I, 2010, Springer-Verlag, Berlin, Heidelberg
34. G.Kousiouris, G. Vafiadis, M. Corrales, "A Cloud Provider Description Schema for meeting legal requirements in cloud federation scenarios", 12th IFIP Conference on e-Business, e-Services, e-Society "Collaborative, trusted and privacy aware e/m-services" I3E 2013, Springer Berlin Heidelberg
35. Agid, "Cloud for Europe Tender: Realization of a Research and Development Project (Pre-Commercial Procurement) on "Cloud For Europe", http://www.agid.gov.it/cloudforeurope
36. Agid, "Annex IV (B) Technical Specification: federated certified service brokerage of EU public administration cloud" http://www.agid.gov.it/sites/default/files/documentazione/annex_iv_b_-_federated_certified_service_brokerage_v103_publish_0_0.pdf