A Knowledge Base Driven Solution for Smart Cloud Management

P. Bellini, D. Cenni, P. Nesi

New York City, USA
June 27 - July 2, 2015
11th IEEE World Congress on Services

The complexity of Cloud infrastructures is increasing every year, requiring new concepts and tools to face off topics such as:

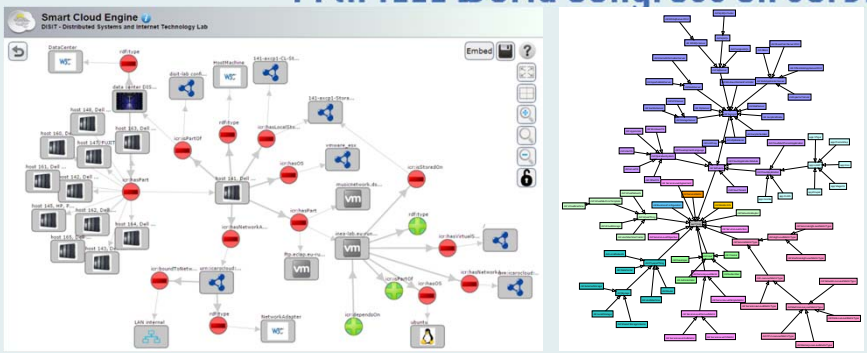
- process configuration and reconfiguration,
- automatic scaling,
- elastic computing and healthiness control.

This paper presents a Smart Cloud solution based on a **Knowledge Base**, with the aim of modeling **cloud resources, Service Level Agreements** and their evolution, using our *Cloud Ontology*, while enabling the reasoning on cloud structures and implementing strategies of efficient smart cloud management and intelligence.

The solution proposed is composed of

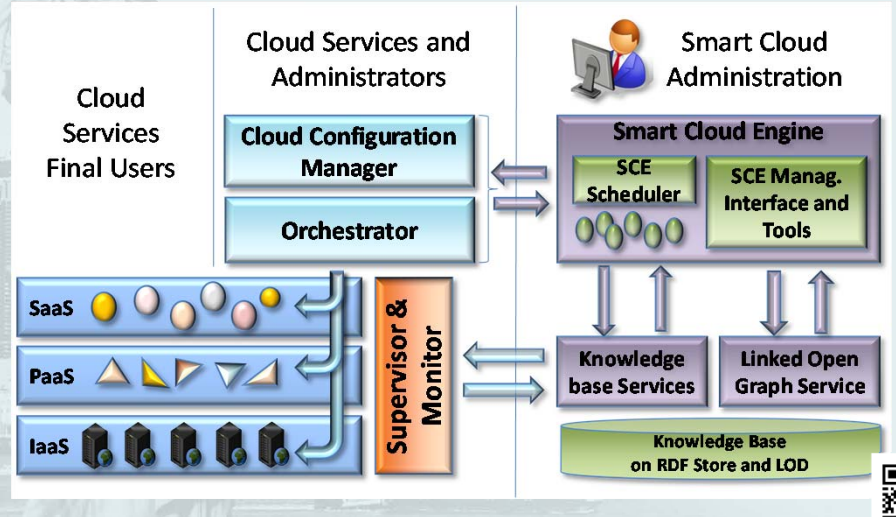
- **Smart Cloud Engine**, SCE
- **Knowledge Base**, KB
- **Supervisor and Monitor** module for data acquisition.

It can be easily integrated with any cloud configuration manager, cloud orchestrator, and monitoring tool, since the connections with these tools are performed by using REST calls and XML files.



Cloud Knowledge Graph & Cloud Ontology

http://log.disit.org **http://www.disit.org/cloud_ontology/core**



DISIT Smart Cloud Architecture

Validation & Verification
Specific SPARQL queries over the KB are used to validate the configurations submitted, checking the formal structure and also the feasibility of the new configuration.

Smart Cloud Engine
periodically checks the status of the resources in the cloud infrastructure (e.g., virtual machines and application services), connects to the Knowledge Base through the use of SPARQL queries, and invokes appropriate REST calls toward the Configuration Manager (CM), as defined in the Service Level Agreement related to the specific cloud service of interest (e.g., scaling, balancing, reconfiguration).

Experiments & Validation
The most complex SLA has 75 conditions for an application (www.eclap.eu, a social network with scalable frontend and backend) using 13 VMs and running 12 services (1 HTTP balancer, 3 Web Servers, 1 Apache Tomcat, 1 MySQL, 1 AXCP Scheduler, 5 AXCP Grid Nodes). With an history of service metric values of about 3 months (with about 3800 measures per metric), the time needed to evaluate a SPARQL query to evaluate the SLA and to get the current values for the metrics involved is of about 30s, while for a SLA on a single VM with four conditions (with bounds on CPU usage percentage, memory, disk storage and network metrics) it takes about 2.0 s.

Paolo Nesi, University of Florence, Italy
paolo.nesi@unifi.it
http://www.disit.dinfo.unifi.it