TRADING COMPLEX SERVICES ON CLOUD-ENABLED SERVICE PLATFORMS

FLEXCLOUD: FLEXIBLE SERVICE ARCHITECTURES FOR CLOUD COMPUTING

The Internet of Services visions inspire to lift concepts from the conventional notion of service orientation towards a broader service definition. All services shall be uniformly described, brokered, and managed throughout a complete lifecycle with provider and consumer perspectives. The *SPACE-Cloud* demonstrator shows how complex services in a cloud fit into this uniformity model.

Josef Spillner, Ronny Seiger, Johannes Schad, Stephan Zepezauer, Alexander Schill

INTRODUCTION

Today, the number and variety providers of cloud computing is on the rise. However, the manageability of the individual offerings is poor, especially for combined management of all services of a user. A promising approach to keep the overview over offered and actually used services results from recent Internet of Services visions[1]. It promotes richer descriptions on user-centric marketplaces.

The open-source Service Platform Architecture for Contracting and Execution (SPACE) [3] implements core features for service trading along the service usage lifecycle: Offer, search, configuration, use and rating. The design of SPACE mandates modularity by separating the functionality into platform services, and extensibility through integration with these services [2]. SPACE also includes abstraction layers to shield the platform from evolutions in service artefacts such as descriptions and implementations. The service hosting component, which implements the concepts of Unified Service Hosting [1], is the part of SPACE in which heterogeneous implementation technologies are bound to a unified interface for service deployment, execution and monitoring.

ARCHITECTURE AND IMPLEMENTATION

The *SPACE-Cloud* design assumes a cloud platform with programmatic remote access to its functions, for instance through web service interfaces. The cloud itself would be hidden behind the appropriate SPACE platform services and not be exposed to the user directly. Consequently, trading complex services requires several extension points in SPACE:

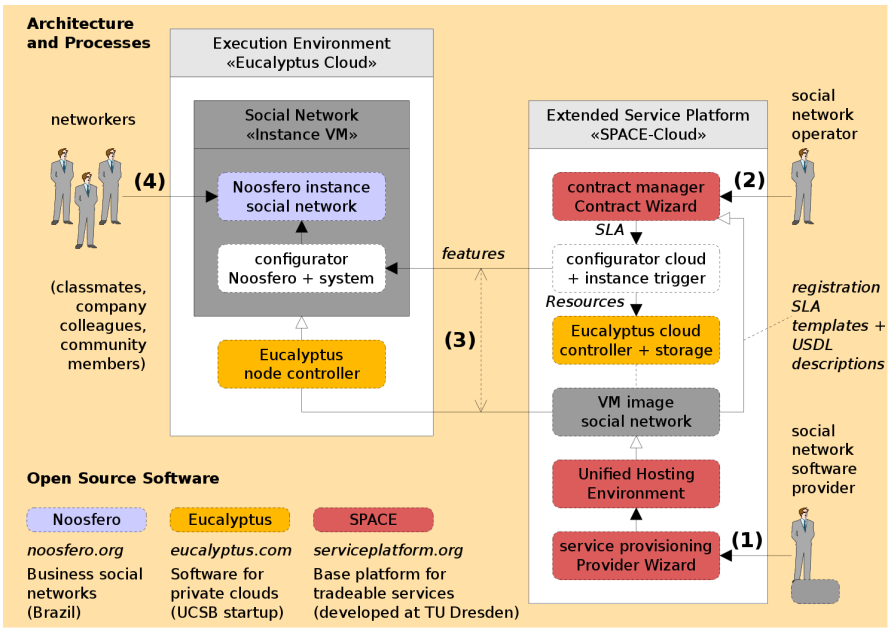

Figure 1: Architecture of *SPACE-Cloud* including a complex scenario service

Our demonstrator shows that by integrating a cloud platform, trading of complex services becomes possible purely by extending the hosting part with negligible modifications of the platform itself. As desired effect, the uniform definition of technical services is extended to beyond the language framework and process level (e.g., Java Axis services and BPEL) to include system-level virtual machines containing multiple complex components. This brings the Internet of Services visions to cloud users.

Cloud Initialisation. Before the cloud can be attached to SPACE, it needs to be initialised with a certain pool of resources, including hard disk space, memory, CPUs and network-related ones.

Service Deployment. When a complex service image is deployed, the service descriptions and agreement templates are extracted from it before the image is pushed into to the cloud image store.

[1] Due to the lack of formal IoS definitions, please refer to THESEUS/TEXO at `http://www.theseus-programm.de` and to the IoS community page `http://www.internet-of-services.com`

Prof. A. Schill | +49 351 463-38263 | alexander.schill@tu-dresden.de | Nöthnitzer Straße 46 | 01062 Dresden | Germany
Ansprechpartner: Dr.-Ing. Stephan Groß | +49 351 463-38213 | stephan.gross@tu-dresden.de | http://flexcloud.eu

TECHNISCHE UNIVERSITÄT DRESDEN
FlexCloud
Europa fördert Sachsen. ESF Europäischer Sozialfonds SACHSEN

Contract Negotiation. When a service level agreement is negotiated between the provider and the prospective consumer, the resource demand is calculated and the cloud is instructed to allocate sufficient resources.
Eucalyptus was chosen as cloud management stack as it reimplements the well-known Amazon cloud[2]. The scenario virtual machine contains a minimal Linux operating system, a Ruby-on-Rails runtime, an unconfigured social network framework[3], as well as an auxiliary directory used to store declarative service artefacts. The flow of requests and data through the resulting architecture is shown in Fig. 1. The diagram combines service deployment (1) with contract negotiation (2) and subsequent service instantiation (3).

RESULTS

SPACE-Cloud has been integrated into SPACEflight, a live demonstrator combining SPACE with an operating system, a custom desktop with service frontend delivery, and scenario services. The desktop with its cloud-related deep-dive bookmarks can be seen in Fig. 2.

Figure 2: Welcome desktop of the SPACEflight demonstrator

Fig. 3 shows the service discovery process which is based on semantic descriptions with non-functional property specifications. In the cloud scenario service, the descriptions encompass properties which indirectly affect the resource allocation, such as the number of expected users in the social network.

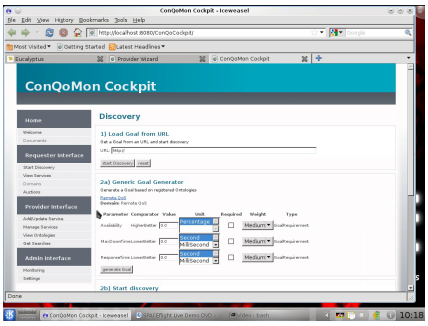

Figure 3: Discovery of matching services with certain non-functional properties

The administrator interface of Eucalyptus in Fig. 4 shows the initial cloud configuration as well as all deployed service images and custom instantiation profiles bound to service level agreements.

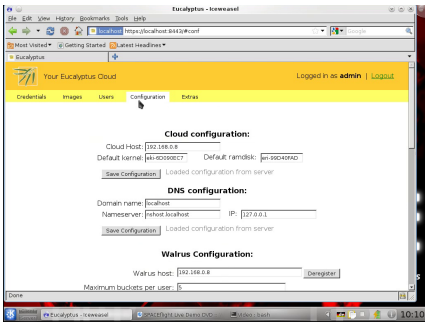

Figure 4: Deployment of complex services into Eucalyptus

Several lessons could be learned from the development of *SPACE-Cloud*. When service packages become as large as typical virtual machine images, the web service interfaces for deployment should offer asynchronous operation with a feedback channel on the operation progress. Similarly, zero-copy techniques should be used to avoid the multiplication of the image size due to the original location, the deployment temporary files, the cloud image store and caches.

CONCLUSION

With *SPACE-Cloud*, we have successfully extended the SPACE platform towards a broader technical service definition. Complex services on the virtual machine level can be deployed alongside more lightweight business processes and atomic services under a unified hosting model. Our contribution intends to accelerate the development of a new service economy on the Internet in which users and companies offer and consume services transparently over independent marketplaces.
The resulting demonstrator SPACEflight 1.0 beta5 has already been shown at CeBIT 2011 and is publicly available for download and evaluation[4].

REFERENCES

[1] J. Spillner, I. Braun, and A. Schill. Towards Unified Service Hosting. In *INSTICC-Tagungsband: 4th International Conference on Software and Data Technologies*, volume 2, pages 31–36, July 2009. Sofia, Bulgaria.

[2] J. Spillner, A. Kümpel, I. Braun, and A. Schill. Infrastruktur zur experimentellen Evaluierung von Konzepten im Internet der Dienste. In *Workshop Internet der Dienste (IDD) auf der 40. GI-Jahrestagung INFORMATIK*, September 2010. Leipzig, Germany.

[3] J. Spillner and A. Schill. The TECJIA Service Platform: Web Service Sharing based on Modular Platform Services. In *Proceedings of FIS - 2nd Future Internet Symposium*, September 2009. Berlin, Germany.

[2] Eucalyptus cloud management: `http://open.eucalyptus.com`
[3] Noosfero social network: `http://www.noosfero.org`
[4] SPACEflight download: `http://cloud.serviceplatform.org`

Prof. A. Schill | +49 351 463-38263 | alexander.schill@tu-dresden.de | Nöthnitzer Straße 46 | 01062 Dresden | Germany
Ansprechpartner: Dr.-Ing. Stephan Groß | +49 351 463-38213 | stephan.gross@tu-dresden.de | http://flexcloud.eu