

Will Be Defined After the Brainstorming Phase^{*}

Eirik Brandztæg^{1,2} and Sébastien Mosser¹

¹ SINTEF IKT, Oslo, Norway

² University of Oslo, Oslo, Norway
{firstname.lastname}@sintef.no

Abstract. ~150 words expected. Will also be defined after the brainstorming phase. Must mention *(i)*the problem, *(ii)*the actual contribution and *(iii)*the obtained results.

1 Introduction

I'll write the introduction afterwards, when the content of the paper will be fixed.

- Cloud-computing research field [3]
- Model-driven engineering applied to the cloud

2 Challenges in the cloud

]DMTF.OpenVirtualizationFormatSpec Technical Report DSP0243, DMTF, 2009.
] Eclipse. <http://eclipse.org>, 2009.

To recognize cloud provisioning challenges an example application[4] has been utilized. The application (from here known as BankManager) is a featureless bank manager system written in Grails[7], it supports creating users and bank accounts, moving money between bank accounts and users. BankManager is designed to be distributed between several nodes with two front-end applications connected to one back-end database as seen in Figure 1 with three nodes and one browser to visualize application flow. To prototype deployment of the design scripts were used to create instances and deploy the software. From this prototype it became clear that there were several challenges that CloudML must tackle:

- **Dependency data between instances:** Some nodes need to know about other nodes in runtime. Some of this data might not be available until dependant nodes reach a running state
- **Technical competence:** The level of knowledge needed to manually provision nodes is more technical and not business liable. Even more so for multicloud provisioning

^{*} This work is funded by the European commission through the REMICS project, contract number 257793, with the 7th Framework Program.

- **Reusability:** Once a cluster of nodes are manually initialized it is difficult to replicate the setup on other clouds, or even on the same provider
- **Robustness:** There were several ways the scripts could fail and most errors were ignored
- **Complexity:** There are some standards[6] for clouds and provisioning on cloud environments. But the providers we tested against[2] did not behave equally, and the APIs were different.

Use tikz instead?

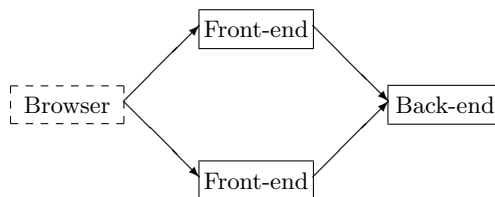


Fig. 1. Nodes for deploying BankManager

3 Contribution

CloudML is a lexical model-based language for cloud provisioning designed to solve the challenges presented in SEC. 2. The language has two main models (Account and Template) as seen in FIG. 2, one used for provider credentials and the other to define nodes for provisioning. The two models use the same lexical syntax but are separated in the figure because combining them is semantically wrong, as they are only used together during runtime without any link to runtime models. The template model has similarities to a *Virtual Deployment Model* as described by Konstantinou[1] although it also faces the challenge of simplicity. This model-based provisioning provide underlying benefits that tackles some of the challenges.

Implementation. The design is implemented as a proof of concept framework[5] (from here known as *cloudml-engine*). Cloudml-engine is written in Scala, builds with Maven and can be used from Java. The templates from the design are constructed with JavaScript Object Notation (JSON). Clouml-engine support use jclouds.org library to connect with cloud provider, giving it support for several providers out of the box, but the main focus is towards Amazon AWS and Rackspace Cloudservers.

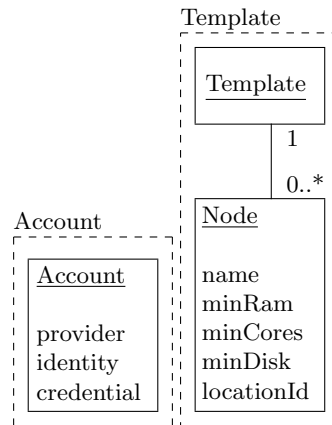


Fig. 2. Model of Account and Template

4 Validation & Experiments

- Three instances
 1. Frontend webapp
 2. Frontend webapp
 3. Backend database
- Setup works on
 - AWS EC2
 - Rackspace cloudservers

5 Related Works

- AWS CloudFormation (Amazon only)
- jclouds (Only through (more advance?) code)
- libcloud (Only through (more advance?) code)
- CA Applogic (only graphical, and inhouse)

6 Conclusions

I'll write the conclusions afterwards.

References

1. A. V. Konstantinou, T. Eilam, M.K.A.A.T.W.A., E.Snible: An architecture for virtual solution composition and deployment in infrastructure clouds. Tech. rep., IBM Research (2009)
2. Amazon: Amazon web services (2012), <http://aws.amazon.com/>

3. Armbrust, M., Fox, A., Griffith, R., Joseph, A.D., Katz, R.H., Konwinski, A., Lee, G., Patterson, D.A., Rabkin, A., Stoica, I., Zaharia, M.: Above the Clouds: A Berkeley View of Cloud Computing. Tech. Rep. UCB/EECS-2009-28, EECS Department, University of California, Berkeley (Feb 2009), <http://www.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-28.html>
4. Brandtze, E.: Bank manager (2012), <https://github.com/eirikb/grails-bank-example>
5. Brandtze, E.: cloudml-engine (2012), <https://github.com/eirikb/cloudml-engine>
6. Force, D.M.T.: Open virtualization format (2012), <http://www.dmtf.org/standards/ovf>
7. SpringSource: Grails (2012), <http://grails.org>