The CoCoME Platform for Collaborative Empirical Research on Information System Evolution

Institute for Program Structures and Data Organization

Table of Contents

1 Introduction

2 CoCoME Plattform Overview

3 Evolution Subject

In this paper, we use CoCoME as evolution subject. CoCoME has been further evolved in various projects, using the existing Hybrid Cloud-based Variant and a new approach called Microservice-based Variant. The former is sufficiently presented in the official Technical Report of CoCoME. See [Hein16] for any information and implementation details.

3.1 Hybrid Cloud-based Variant

The hybrid cloud-based variant of CoCoME was developed in the DFG Priority Programme Design For Future - Managed Software Evolution (SPP 1593) [GRGH+15]. This variant of CoCoME is using Java EE technologies for both the frontend and the backend. The hybrid cloud-based variant used in this report is completely identical to the variant described in [Hein16].

3.2 Microservice-based Variant

vgl zu 2016,2 Since.. various research projects ggf aendern? mit 3.1, 3.2, 3.3 gemeint ->?

4 Evolution Scenarios

We implemented distinct evolution scenarios covering the categories adaptive and perfective evolution. Corrective evolution is not considered in the scenarios as this merely refers to fixing design or implementation issues.

4.1 Evolution Scenarios of the Hybrid Cloud-based Variant

This section introduces the two evolution scenarios of the hybrid cloud-based variant of CoCoME.

4.1.1 Setting up a Docker environment

The CoCoME company must reduce IT administration costs but frequent updates to the enterprise and store software are necessary to continuously improve the entire system. As a consequence, IT staff need to update the system components as soon as a new software version is released. An Operations Team member has to get access to the actual server in order to undeploy the old version and replace it with the new one. This is time consuming and expensive as the updates have to be done manually.
Therefore, a Docker version is elaborated to simplify the administration process. As soon as a new software version of CoCoME is ready for delivery, the Development Team wrap it into a Docker Image. This Image can be automatically deployed to the destination server according to the principle of Continuous Deployment (CD) [OlAB12].

4.1.2 Adding a Mobile App

After successfully adding a Pick-up Shop, the CoCoME company stays competitive with other online shop vendors (such as Amazon). In times of smartphones, customer do not only want to buy exclusively goods from their home computers. Purchasing goods 'on the way' comes more and more into fashion. This raises the idea to create a second sales channel next to the existing Pick-up Shop in the CoCoME system. As a consequence, more customers can be attracted to gain a larger share of the market.
The customer can order and pay by using the app. The delivery process is similar to the Pick-up Shop: The goods are delivered to a pick-up place (i.e. a store) of her/his choice, for example in the neighbourhood or the way to work. By introducing the Mobile App as a multi OS application, the CoCoME system has to face various quality issues such as privacy, security and reliability. Also the performance of the whole application can be affected if many customers order via the app.

4.2 Evolution Scenarios of the Microservice-based Variant

This section introduces the evolution scenario of the Microservice-based variant of CoCoME.

4.2.1 Defining different Microservices

5 Design Details for Evolution Scenarios

In this chapter we provide the detailed design documentation for each of the evolution scenarios introduced in the prior section. Sec. 5.1 sketches the design decision for the Mobile App that provides a second sales channel next to the existing Pick-up Shop. Sec. 5.2 describes the adaptive evolution scenario of setting up a Docker environment to simplify the update process. They are both based on, or at least use the Hybrid Cloud-based Variant of CoCoME [Hein16]. In contrast, Sec. 5.3 provides a detailed design documentation of a new architectural version of CoCoME. This perfective evolution scenario is realized based on the Microservice idea.

5.1 Design Decisions for the Mobile App

5.2 Setting up a Docker environment

5.3 Using Microservices Technology

- je microservice absatz mit entsprechenden Sequenzendiagram
- frontend? muss dazu auch das gemacht werden?
- ein blocktext zu mehereren diagrammen oder diagramme zwischen text?
- je die einzelnen module und szenarien erlaeutern in dem diese sinnvoll sind

5.3.1 **Products** abstrahieren der Produktinformationen

5.3.2 **Stores** einzelne Laeden alleinstehend abbilden um nach bedarf neue microservices alias laeden starten zu kÃ¶nnen

5.3.3 **Enterprise** aehnlich zu stroes

5.3.4 **Reports** stellt alleinigen aufgaben bereich dar, entsprechen undabhaengig darzustellen von anderem.

(grafik/Tabelle -> oben oder unten caption)

6 Life-Cycle

was genau ist hiermit gemeint

7 Implementation of Evolution Scenarios

gemeint die abh. zwischen den einzelnen klassen? - bei mobile app angegeben, alt. wo ist der quellcode?
- docker den techstack (nocheinmal?) angeben? - microservices hierzu komplett fertig? oder nur die einzelnen mittelstÃ¼cke? ohne entsprechende anbindung an frontend/backend?

8 Conclusion

References

GRGH+15. Ursula Goltz, Ralf H. Reussner, Michael Goedicke, Wilhelm Hasselbring, Lukas Märtin und Birgit Vogel-Heuser. Design for future: managed software evolution. *Computer Science - Research and Development*, 30(3), Aug 2015, S. 321–331.

Hein16. Robert [VerfasserIn] Heinrich. The CoCoME platform for collaborative empirical research on information system evolution, [2016].

OlAB12. Helena Holmström Olsson, Hiva Alahyari und Jan Bosch. Climbing the" Stairway to Heaven"–A Mulitiple-Case Study Exploring Barriers in the Transition from Agile Development towards Continuous Deployment of Software. In *Software Engineering and Advanced Applications (SEAA), 2012 38th EUROMICRO Conference on*. IEEE, 2012, S. 392–399.