

# Projektbericht

► Project Note

## SOPHIST GmbH

Vordere Cramergasse 13  
90478 Nürnberg  
Deutschland

Konzeption und  
Realisierung einer  
Cloud-Verwaltung

# Projektbericht

► Project Note

## SOPHIST GmbH

Vordere Cramergasse 13  
90478 Nürnberg  
Deutschland

Konzeption und  
Realisierung einer  
Cloud-Verwaltung

# Konzeption und Realisierung einer Cloud-Verwaltung

## Aufgabe:

Die ProfitBricks GmbH stellt ihren Kunden physikalische Ressourcen, wie Rechner, Speicher, IP-Adressen etc. zur Verfügung. Um diese in Anspruch zu nehmen, definieren die Kunden eine virtuelle Rechnerumgebung, die dann über eine Schnittstelle an ein Verwaltungstool übergeben wird. Die Aufgabe des Verwaltungstools besteht dann darin, die angeforderten, virtuellen Ressourcen möglichst optimal auf die physikalischen Ressourcen abzubilden, und danach die entsprechenden Kommandos für die Umsetzung in der Physik zu generieren. Aufgrund teilweise langer Ausführungszeiten kann es dabei zu Konflikten kommen, die entsprechend erkannt und gelöst werden müssen. Weitere, wichtige Qualitätsanforderungen waren die Ausfallsicherheit und Fehlertoleranz. Eine weitere Herausforderung war die Definition der Schnittstellen, gerade zu der Physik und zu einem internen Billing-System, da diese im Rahmen dieses Projekts in Absprache mit den Verantwortlichen angepasst werden sollte. Unter anderem deswegen sollten in diesem Projekt die Analyse- und Architekturphasen sehr detailliert gestaltet werden.

## Umsetzung:

Zunächst begannen wir mit einer klassischen Use-Case-Analyse, da zunächst die Funktionalitäten einer Web-Oberfläche für die Anwender mit betrachtet werden sollten. Da dies jedoch zu viele Änderungen bedeutet hätte, einigte man sich auf die Betrachtung der Schnittstelle zu dieser Web-Oberfläche, an der nur die nötigsten Änderungen vorgenommen werden sollten. Mit diesem Aufsatzpunkt konnte in die detaillierte Analyse der internen Funktionalitäten des Verwaltungstools begonnen werden. Hier wurde mit der Modellierung eines Begriffsmodells für die fachlichen Daten begonnen. Damit war die Basis für die Modellierung der internen Systemabläufe gelegt. Ein Grund für die Fokussierung auf die internen Abläufe war, dass hier das meiste Wissen einfließen musste; Anforderungen auf Black-Box-Ebene wären zu simpel für die Aufgabenstellung gewesen. Bei der Modellierung der internen Abläufe haben wir zum Teil auf einige, sehr grobe Architekturentscheidungen aufgesetzt, um die sehr komplexen Abläufe voneinander abgrenzen zu können.

Trotzdem mussten wir mehrere Lösungen ausprobieren, um die Anfragen von der Web-Oberfläche so zu abzuwickeln, dass die Qualitätsanforderungen (insbesondere die Fehlertoleranz) erfüllt wurden. Zum Schluss waren die wichtigsten und komplexesten Abläufe in ca. 50 Aktivitätsdiagrammen modelliert. Nachdem die wichtigsten Analyseergebnisse feststanden, haben wir mit den Architekturbetrachtungen begonnen. Hierbei stand im Vordergrund die Verwendung eines Frameworks, das sowohl die Persistierung als auch eine sehr asynchrone Kommunikation innerhalb des Systems festlegte. Als Vorteil stellte sich bei der Verwendung dieses Frameworks heraus, dass dafür viele benötigte Funktionalitäten generiert werden konnten. Aufsetzend auf das Begriffsmodell aus der Analyse wurden die Implementierungsklassen modelliert und mit verschiedenen Stereotypen versehen, um eine angepasste Generierung zu erlauben. Darüber hinaus wurden in der Architektur die benötigten Komponenten modelliert und ihre Aufgaben durch eine Traceability zu den Analyseergebnissen festgelegt. Zur Identifikation der Komponenten und zur Veranschaulichung der Realisierung wurden die Datenflüsse zwischen diesen Komponenten bzgl. einiger großer Funktionalitäten aus der Analysephase dargestellt. Für immer wiederkehrende Aufgaben in der Implementierung wurden einige Sequenzdiagramme als eine Art Pattern modelliert.

### Kundennutzen:

- ▶ Eine regelgeleitete Analyse machte bewusst, an welchen Stellen eine detailliertere Analyse notwendig war
- ▶ Die Darstellung in UML ermöglichte die Bewertung der Lösungen bzgl. kritischer Szenarien
- ▶ Die Schnittstellen zu Nachbarsystemen war eindeutig in UML identifizierbar
- ▶ Es entstand eine durchgängige Darstellung der Aufgaben des Systems und eine grobe Sicht der Umsetzung
- ▶ Die eingesetzte Methodik hat das Potenzial, auch in anderen Bereichen der Organisation eingesetzt zu werden.



## TECHNISCHE DATEN:

**Eingesetzte Tools:**  
Enterprise Architect

**Eingesetzte Methoden:**  
SOPHIST-REgelwerk zur Identifikation weiterer benötigter Spezifikationsinhalte  
Hierarchische Modellierung der systeminternen Abläufe mit Aktivitätsdiagrammen

---

SOPHIST GmbH  
Vordere Cramergasse 13  
90478 Nürnberg  
Deutschland

fon: +49 (0)9 11 40 900-0  
fax: +49 (0)9 11 40 900-99

E-Mail: [heureka@sophist.de](mailto:heureka@sophist.de)  
Internet: [www.sophist.de](http://www.sophist.de)  
© SOPHIST