Anforderungsspezifikation

Silvan Adrian
Fabian Binna

1 Änderungshistorie

Datum	Version	Änderung	Autor
02.10.15	1.00	Erstellung des Dokuments	Gruppe
02.10.15	1.01	Nicht funktionale Anforderungen	Silvan Adrian
02.10.15	1.02	Use Cases Aktoren + User Stories Aktoren	Silvan Adrian

Inhaltsverzeichnis

2 Einführung

2.1 Zweck

Dieses Dokument beinhaltet die Anforderung zur Analyse.

2.2 Gültigkeitsbereich

Dieses Dokument ist während des ganzen Projekts gültig.

2.3 Referenzen

-

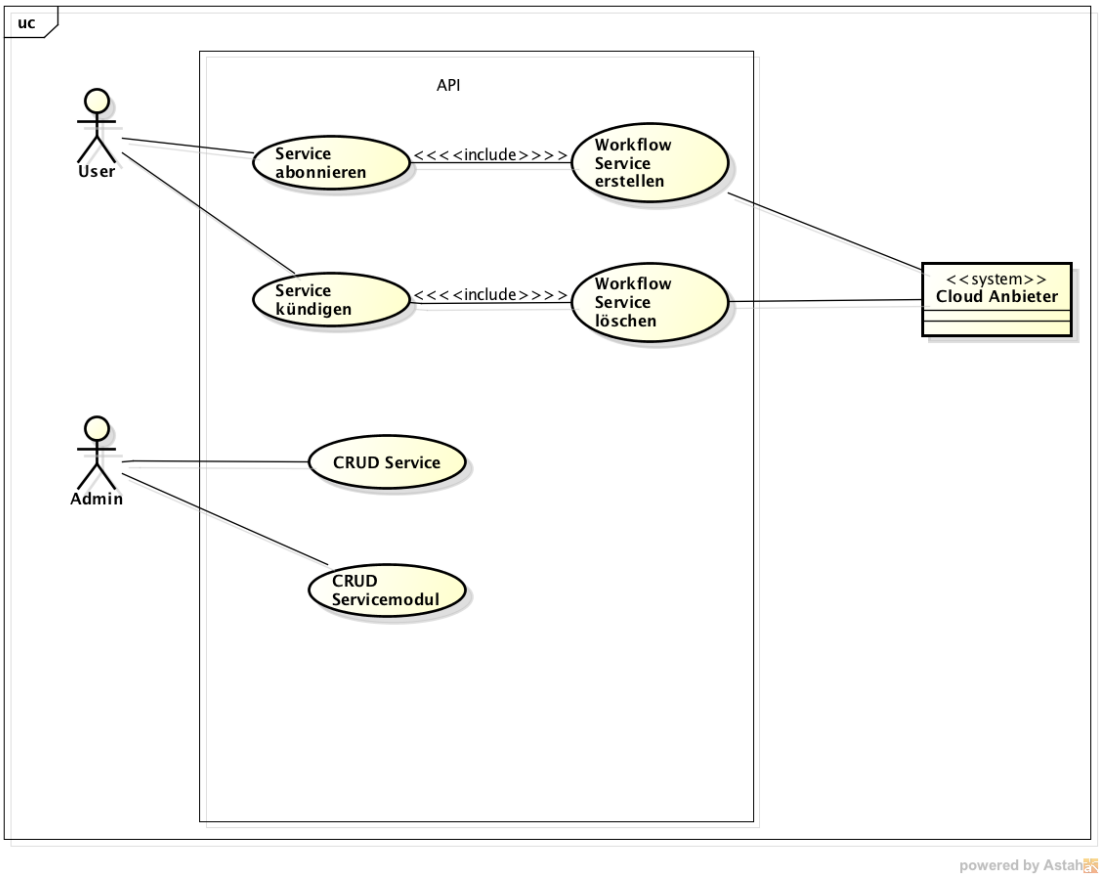
3 Anforderungen

3.1 API

3.2 Dashboard

4 Use Cases

4.1 Use Case Diagramm



4.2 Aktoren & Stakeholders

4.2.1 User

Als User möchte ich meine abonnierten Services verwalten.

Aktor	Typ	Ziele
User	Primary	• Service abonnieren • Service kündigen

4.2.2 Admin

Als Admin möchte ich Services und Servicemodule verwalten können.

Aktor	Typ	Ziele
Admin	Primary	• Service erstellen • Service anpassen • Service löschen • Servicemodul erstellen • Servicemodul anpassen • Servicemodul löschen

4.3 Beschreibungen fully dressed

4.3.1 Service abonnieren

Primäraktor	User
Steakholders und Interessen	Spieler: Möchte einen Service abonnieren
Vorbedingungen	Das User-Dashboard wurde geöffnet
Nachbedingungen	Das User-Dashboard wurde geschlossen
Standartablauf	1. Der User gibt die Webadresse für das Dashboard ein
Spezielle Anforderungen	siehe nichtfunktionale Anforderungen
Technologie- und Datenvarianten	Keine
Auftrittshäufigkeit	mehrmals pro Woche
Offene Fragen	Keine

5 User Stories

5.1 Rollen

5.1.1 User

Als User benutze ich das Dashboard, um mir einen Service zu abonnieren und

5.1.2 Admin

Als Admin benutze ich die API über die Kommandozeile oder nutze das Admin-Dashboard um neue Services zusammenzustellen.

6 Nichtfunktionale Anforderungen

6.1 Menge

- Die Software unterstützt mehr als 30 Cloud Anbieter (libcloud)
- Bei jedem Cloud Anbieter bestehen eine gewisse Anzahl Services (von Anbieter zu Anbieter verschieden)

6.2 Schnittstellen

- Die Software wird über HTTP/HTTPS angesprochen
- Zur Interaktion im Admin-Dashboard werden die herkömmlichen Schnittstellen gebraucht (Maus,Tastatur,Bildschirm)
- Interaktionen können auch über die Kommandozeile ausgeführt werden

6.3 Qualitätsmerkmale

6.3.1 Funktionalität

siehe Abschnitt API und Dashboard

6.3.2 Zuverlässigkeit

- Der Workflow zum erstellen eines Services soll entweder durchgeführt und abgeschlossen werden oder falls Unterbruch/Fehler rückgängig gemacht werden.
- Die Software soll verteilt betrieben werden und eine möglichst hohe Verfügbarkeit bieten

6.3.3 Benutzerbarkeit

- Die Software kann über das vorgesehene Admin-Dashboard benutzt werden
- Die API kann auch über die Kommandozeile angesprochen werden

6.3.4 Effizienz

-

6.3.5 Änderbarkeit

Die Software soll modular aufgebaut werden, damit Erweiterungen in Zukunft problemlos möglich sind.

6.3.6 Übertragbarkeit

Das Projekt wird in Python geschrieben ist somit also auf Python mindestens in der Version 2.5 angewiesen, kann allerdings durch den Einsatz eines Docker Containers einfach Übertragbar gemacht werden.