

# SDDC

**Software Defined Data Center**

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
03.10.15	1.03	Anforderungen API	Fabian Binna
03.10.15	1.04	Anforderungen Dashboards + Mockups eingefügt	Silvan Adrian
03.10.15	1.05	Use Cases fully dressed	Silvan Adrian

# Inhaltsverzeichnis

<b>1</b>	<b>Änderungshistorie</b>	<b>2</b>
<b>2</b>	<b>Einführung</b>	<b>5</b>
2.1	Zweck . . . . .	5
2.2	Gültigkeitsbereich . . . . .	5
2.3	Referenzen . . . . .	5
<b>3</b>	<b>Anforderungen</b>	<b>5</b>
3.1	API . . . . .	5
3.2	Customer-Dashboard . . . . .	6
3.2.1	Homescreen . . . . .	6
3.2.2	Services Übersicht . . . . .	6
3.2.3	Service abonnieren . . . . .	6
3.2.4	Cloud Credentials . . . . .	7
3.3	Admin-Dashboard . . . . .	7
3.3.1	Service . . . . .	7
3.3.2	Servicemodul . . . . .	8
<b>4</b>	<b>Use Cases</b>	<b>9</b>
4.1	Use Case Diagramm . . . . .	9
4.2	Aktoren & Stakeholders . . . . .	9
4.2.1	Customer . . . . .	9
4.2.2	Admin . . . . .	10
4.3	Beschreibungen fully dressed . . . . .	11
4.3.1	UC01: Service abonnieren . . . . .	11
4.3.2	UC02: Service kündigen . . . . .	13
4.3.3	UC03: Cloud Logindaten verwalten . . . . .	15
4.3.4	UC04: Services verwalten . . . . .	16
4.3.5	UC05: Servicemodule verwalten . . . . .	18
<b>5</b>	<b>User Stories</b>	<b>20</b>
5.1	Rollen . . . . .	20
5.1.1	User . . . . .	20
5.1.2	Admin . . . . .	20
<b>6</b>	<b>Nichtfunktionale Anforderungen</b>	<b>20</b>
6.1	Menge . . . . .	20
6.2	Schnittstellen . . . . .	20
6.3	Qualitätsmerkmale . . . . .	20
6.3.1	Funktionalität . . . . .	20
6.3.2	Zuverlässigkeit . . . . .	20
6.3.3	Benutzerbarkeit . . . . .	20

6.3.4	Effizienz . . . . .	21
6.3.5	Änderbarkeit . . . . .	21
6.3.6	Übertragbarkeit . . . . .	21

## **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**

Die API definiert einen Workflow der einen Service auf einer Cloud erstellt. Es ist offen, ob dieser Service über mehrere Cloud Anbieter hinaus geht. Der Service wird durch ein Konfigurationsfile (z.B. json) definiert. Zusätzlich können Scriptfiles referenziert werden, die die Software auf den Instanzen installieren. Ein Service kann auch wieder gelöscht werden. Es ist nicht die Aufgabe der API existierende Services zu identifizieren. Die API muss Modular sein, das heisst es sollte möglich sein andere oder eigene Programme für die Cloud Kommunikation zu verwenden. Innerhalb der API werden Compute, Storage, Network usw. als ServiceModule bezeichnet. Diese Abstraktion ermöglicht das wiederverwenden und erweitern der API.

## 3.2 Customer-Dashboard

### 3.2.1 Homescreeen

Im Homescreeen (sobald der Customer auf das Dashboard zugreift) werden alle zu Verfügung stehenden Services angezeigt. Hier werden die Services Offerings genannt um eine Unterscheidung zwischen Abonnierten Services (Services) und zur Verfügung stehenden Services (Offerings) machen zu können.

#### Dashboard

#### Offerings Services Account

##### Offerings



### 3.2.2 Services Übersicht

In der Services Übersicht werden dem Customer alle abonnierten Services angezeigt und können hier auch gekündigt werden.

#### Dashboard

#### Offerings Services Account

##### Your Services

Name	Options
LAMP Stack	terminate
MEAN Stack	terminate
<input type="button" value="subscribe Service"/>	

### 3.2.3 Service abonnieren

Sobald ein Service auf dem Homescreeen ausgewählt wird muss noch der zuständige Provider gewählt werden (häng auch wieder davon ab von welchem Provider bisher Logindaten hinterlegt wurden). Da momentan noch kein Hybrid Betrieb vorgesehen ist muss ein Account gewählt werden unter welchem der Service verrechnet wird.

## Dashboard

**Offerings Services Account**

### LAMP Stack

Account

OpenStack

subscribe Service

### 3.2.4 Cloud Credentials

Da für jeden Provider wieder Logindaten benötigt werden müssen diese an einer zentralen Stelle gespeichert werden (anhand von denen kann sich auch das Service Angebot ändern).

## Dashboard

**Offerings Services Account**

[Home](#) > [Account](#) > Cloud Credentials

### Your Cloud Credentials

Name	Provider	Options
silvan	OpenStack	delete / change
sadrian@hswr.ch	Digitalocean	delete / change

add Credentials

## 3.3 Admin-Dashboard

Zusätzlich zum Customer-Dashboard soll ein Admin-Dashboard zur Verfügung stellen in welchem der Admin Services und Servicemodule erstellen kann.

### 3.3.1 Service

Ein Service hat einen bestimmten Namen und jedem Service sind eine gewisse Anzahl Servicemodule zugeteilt, um den Service abbilden zu können. Hier kann der Admin den Service ändern und je nach Anforderung den Service anpassen.

## Dashboard

**Services**  **Servicemodules**  **Account**

### Services

Name	Aktion
LAMP Stack	delete / change
MEAN Stack	delete / change
Build Server	delete / change
<input type="button" value="create Service"/>	

### 3.3.2 Servicemodul

Jedes Servicemodul besitzt einen Namen und wird einem Provider zugeschrieben, dabei kann jedes Servicemodule den Typ Compute,Network oder Storage haben.

## Dashboard

**Services**  **Servicemodules**  **Account**

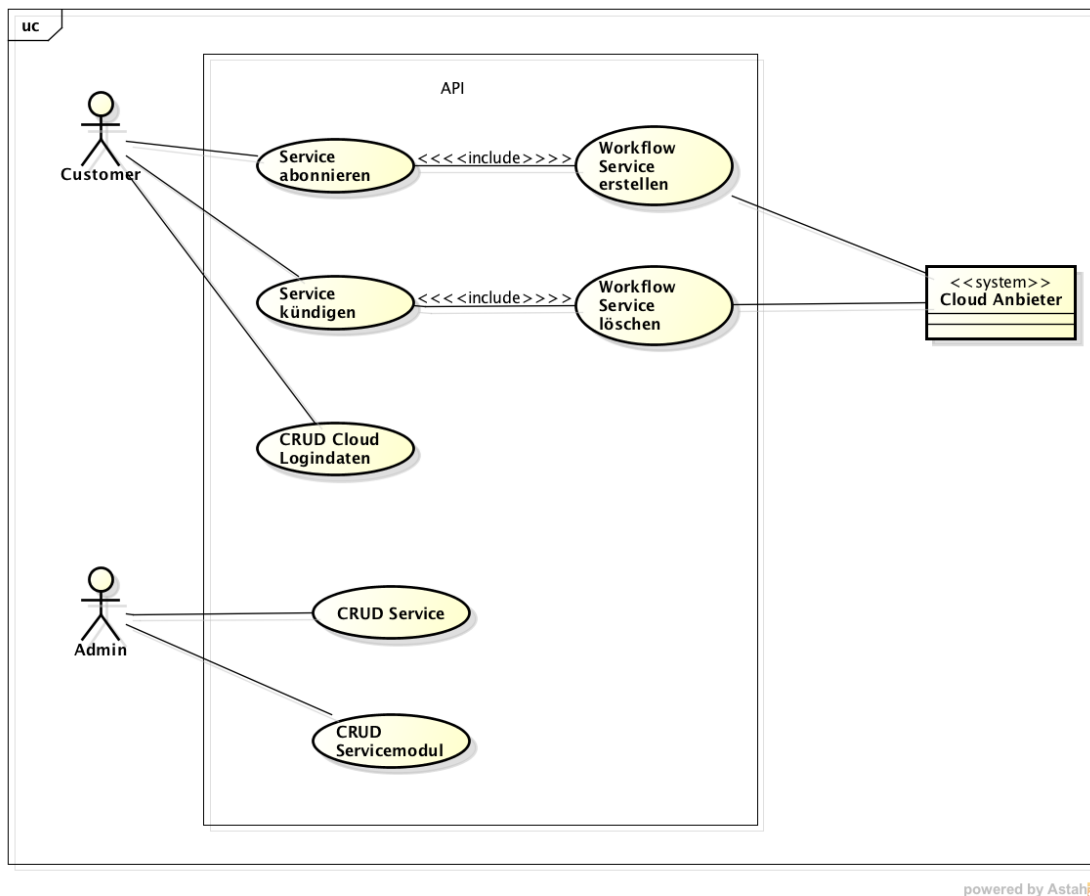
### Servicemodules

Name	Provider	Type	Options
LAMP Instance	OpenStack	Compute	delete / change
MEAN Instance	Azure	Compute	delete / change
Build Server Instance	Cloud Stack	Compute	delete / change
<input type="button" value="create Servicemodule"/>			



## 4 Use Cases

### 4.1 Use Case Diagramm



powered by Astah

### 4.2 Aktoren & Stakeholders

#### 4.2.1 Customer

Als Customer möchte ich meine abonnierten Services verwalten.

Aktor	Typ	Ziele
Customer	Primary	• Service abonnieren
		• Service kündigen
		• Cloud Logindaten hinzufügen
		• Cloud Logindaten anpassen
		• Cloud Logindaten löschen

#### 4.2.2 Admin

Als Admin möchte ich Services und Servicemodule verwalten können.

Aktor	Typ	Ziele
Admin	Primary	<ul style="list-style-type: none"><li>• Service erstellen</li></ul>
		<ul style="list-style-type: none"><li>• Service anpassen</li></ul>
		<ul style="list-style-type: none"><li>• Service löschen</li></ul>
		<ul style="list-style-type: none"><li>• Servicemodul erstellen</li></ul>
		<ul style="list-style-type: none"><li>• Servicemodul anpassen</li></ul>
		<ul style="list-style-type: none"><li>• Servicemodul löschen</li></ul>

---

## 4.3 Beschreibungen fully dressed

### 4.3.1 UC01: Service abonnieren

<b>Primäraktor</b>	Customer
<b>Steakholders und Interessen</b>	Customer: Möchte einen Service abonnieren
<b>Vorbedingungen</b>	Das Customer-Dashboard wurde geöffnet, ist bei der API authentifiziert und hat einen Cloud Account hinzugefügt.
<b>Nachbedingungen</b>	Die Service Infos wurden gespeichert und der Workflow wurde angestossen

#### Standartablauf

1. Der Customer gibt die Webadresse für das Dashboard ein
2. Wiederholen bis kein Service mehr abonniert werden soll
  - a) Der Customer wechselt in die **Offerings Übersicht**
  - b) Der Customer wählt einen der vorhandenen **Services** aus
  - c) Der Customer wählt einen vorhanden **Account** auf welchem er den **Service** abonnieren will
  - d) Der Customer drückt den Button **subscribe Service**
  - e) Der Customer wird in die **Services Übersicht** weitergeleitet
3. Der Customer schliesst das Customer-Dashboard

---

## Alternativer Ablauf

1. a) Der Customer entscheidet sich um und schliesst das Fenster/Tab
2. a) Der Customer entscheidet sich um
  - i. Schliesst das Fenster/Tab
- b) Der Customer entscheidet sich um
  - i. Schliesst das Fenster/Tab
  - ii. geht zurück in die **Offerings Übersicht**
- c) Der Customer entscheidet sich um
  - i. Schliesst das Fenster/Tab
  - ii. geht zurück in die **Offerings Übersicht**
  - iii. wählt einen anderen Account
- d) Der Customer entscheidet sich um
  - i. Schliesst das Fenster/Tab
  - ii. geht zurück in die **Offerings Übersicht**
  - iii. wählt einen anderen Account

---

<b>Spezielle Anforderungen</b>	siehe nichtfunktionale Anforderungen
<b>Technologie- und Datenvarianten</b>	Keine
<b>Auftrittshäufigkeit</b>	mehrmals pro Woche
<b>Offene Fragen</b>	Keine

---

#### 4.3.2 UC02: Service kündigen

<b>Primäraktor</b>	Customer
<b>Steakholders und Interessen</b>	Customer: Möchte einen Service kündigen
<b>Vorbedingungen</b>	Das Customer-Dashboard wurde geöffnet, ist bei der API authentifiziert und hat einen Service abonniert
<b>Nachbedingungen</b>	Die Service Infos wurden gelöscht und der Workflow wurde angestossen
<b>Standartablauf</b>	<ol style="list-style-type: none"> <li>1. Der Customer gibt die Webadresse für das Dashboard ein</li> <li>2. Wiederholen bis kein Service mehr gekündigt werden soll <ol style="list-style-type: none"> <li>a) Der Customer wechselt in die <b>Services Übersicht</b></li> <li>b) Der Customer wählt einen der vorhanden <b>Services</b> aus</li> <li>c) Der Customer drückt auf den link <b>terminate</b></li> <li>d) Der Customer wird in die <b>Services Übersicht</b> weitergeleitet</li> </ol> </li> <li>3. Der Customer schliesst das Customer-Dashboard</li> </ol>

---

**Alternativer Ablauf**

1. a) Der Customer entscheidet sich um und schliesst das Fenster/Tab
  2. a) Der Customer entscheidet sich um
    - i. Schliesst das Fenster/Tabb) Der Customer entscheidet sich um
    - i. Schliesst das Fenster/Tab
    - ii. wählt einen anderen Servicec) Der Customer entscheidet sich um
    - i. Schliesst das Fenster/Tab
    - ii. wählt einen anderen Service
- 

---

<b>Spezielle Anforderungen</b>	siehe nichtfunktionale Anforderungen
<b>Technologie- und Datenvarianten</b>	Keine
<b>Auftrittshäufigkeit</b>	mehrmals pro Woche
<b>Offene Fragen</b>	Keine

---

### 4.3.3 UC03: Cloud Logindaten verwalten

<b>Primäraktor</b>	Customer
<b>Steakholders und Interessen</b>	Customer: Möchte ich Cloud Logindaten verwalten
<b>Vorbedingungen</b>	Das Customer-Dashboard wurde geöffnet
<b>Nachbedingungen</b>	Die Account Infos wurden gespeichert
<b>Standartablauf</b>	<ol style="list-style-type: none"> <li>1. Der Customer gibt die Webadresse für das Dashboard ein</li> <li>2. Der Customer wechselt in die <b>Account/Cloud Credentials Übersicht</b></li> <li>3. Solange wiederholen bis kein Account mehr hinzugefügt werden muss <ol style="list-style-type: none"> <li>a) Customer drückt Button <b>Add Credentials</b></li> <li>b) Customer füllt gefragt Felder aus</li> <li>c) Customer bestätigt Eingaben mit Klick auf <b>Save</b></li> </ol> </li> <li>4. Der Customer schliesst das Customer-Dashboard</li> </ol>
<b>Alternativer Ablauf</b>	<ol style="list-style-type: none"> <li>3. <ol style="list-style-type: none"> <li>a) Wiederholen bis kein Account mehr geändert werden muss <ol style="list-style-type: none"> <li>i. Account auswählen und auf <b>change</b> Link klicken</li> <li>ii. Felder anpassen</li> <li>iii. Änderungen bestätigen mit Klick auf Button <b>Save</b></li> </ol> </li> <li>b) Wiederholen bis kein Account mehr gelöscht werden muss <ol style="list-style-type: none"> <li>i. Account auswählen und auf <b>delete</b> Link klicken</li> </ol> </li> </ol> </li> </ol>
<b>Spezielle Anforderungen</b>	siehe nichtfunktionale Anforderungen

<b>Technologie- und Datenvarianten</b>	Keine
<b>Auftrittshäufigkeit</b>	mehrmals pro Woche
<b>Offene Fragen</b>	Keine

#### 4.3.4 UC04: Services verwalten

<b>Primäraktor</b>	Customer
<b>Steakholders und Interessen</b>	Admin: Möchte einen Service verwalten
<b>Vorbedingungen</b>	Das Admin-Dashboard wurde geöffnet und Admin eingeloggt, falls Service gelöscht werden soll darf kein Customer mehr den Service abonniert haben
<b>Nachbedingungen</b>	Das Admin-Dashboard wurde geschlossen und Änderungen wurden gespeichert

##### Standartablauf

1. Der Admin gibt die Webadresse für das Admin-Dashboard ein
2. Wiederholen bis kein neuer Service hinzugefügt werden muss
  - a) Der Admin wechselt in die **Services Übersicht**
  - b) Der Admin drückt auf den button **create Service**
  - c) Der Admin füllt die benötigten Daten ein (**Name, welche Servicemodule**)
  - d) der Admin bestätigt mit Klick auf Button **Save**



---

## Alternativer Ablauf

2.
  - a) Wiederholen bis kein Service mehr geändert werden muss
    - i. Service auswählen und auf Link **change** klicken
    - ii. Daten ändern
    - iii. Durch Klick auf Button **Save** bestätigen
  - b) Wiederholen bis kein Service mehr gelöscht werden muss
    - i. Service auswählen und Link **löschen** auswählen

---

<b>Spezielle Anforderungen</b>	siehe nichtfunktionale Anforderungen
<b>Technologie- und Datenvarianten</b>	Keine
<b>Auftrittshäufigkeit</b>	mehrmals pro Woche
<b>Offene Fragen</b>	Keine

---

#### 4.3.5 UC05: Servicemodule verwalten

<b>Primäraktor</b>	Customer
<b>Steakholders und Interessen</b>	Admin: Möchte ein Servicemodul verwalten
<b>Vorbedingungen</b>	Das Admin-Dashboard wurde geöffnet und Admin eingeloggt, falls Servicemodul gelöscht werden soll darf kein Service mehr das Servicemodul verwenden.
<b>Nachbedingungen</b>	Das Admin-Dashboard wurde geschlossen und Änderungen wurden gespeichert

##### Standartablauf

1. Der Admin gibt die Webadresse für das Admin-Dashboard ein
2. Wiederholen bis kein neues Servicemodul hinzugefügt werden muss
  - a) Der Admin wechselt in die **Servicemodules Übersicht**
  - b) Der Admin drückt auf den button **create Servicemodule**
  - c) Der Admin füllt die benötigten Daten ein (**Name, Provider, Typ**)
  - d) der Admin bestätigt mit Klick auf Button **Save**

##### Alternativer Ablauf

2.
  - a) Wiederholen bis kein Servicemodul mehr geändert werden muss
    - i. Servicemodul auswählen und auf Link **change** klicken
    - ii. Daten ändern
    - iii. Durch Klick auf Button **Save** bestätigen
  - b) Wiederholen bis kein Servicemodul mehr gelöscht werden muss
    - i. Service auswählen und Link **löschen** auswählen

---

<b>Spezielle Anforderungen</b>	siehe nichtfunktionale Anforderungen
<b>Technologie- und Datenvarianten</b>	Keine
<b>Auftrittshäufigkeit</b>	mehrmals pro Woche
<b>Offene Fragen</b>	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.