

SDDC

Software Defined Data Center

Architektur

Silvan Adrian
Fabian Binna

1 Änderungshistorie

Datum	Version	Änderung	Autor
17.09.15	1.00	Erstellung des Dokuments	Gruppe
18.10.15	1.01	Dokumentaufbau + Logische Sicht	Fabian Binna
20.10.15	1.02	Klassenstrukturen	Fabian Binna
26.10.15	1.03	Klassenstrukturen verbessert + Klassenstruktur genericapi	Fabian Binna
14.11.15	1.04	Datenmodell und Verbesserungen	Silvan Adrian
22.11.15	1.05	Logisch Architektur erneuert	Fabian Binna
22.11.15	1.06	Klassenstruktur services domain	Fabian Binna
22.11.15	1.07	Sequenzdiagramm Service bestellen	Fabian Binna
22.11.15	1.08	Deployment Angepasst	Silvan Adrian
29.11.15	1.09	Architektur verbessert	Silvan Adrian
05.12.15	1.10	Services erweitert	Silvan Adrian
06.12.15	1.11	GenericAPI + Dependency Injection	Fabian Binna

Inhaltsverzeichnis

1	Änderungshistorie	2
1.1	Zweck	4
1.2	Gültigkeitsbereich	4
2	Systemübersicht	5
3	Logische Architektur	5
4	Klassenstruktur	6
4.1	services	6
4.2	domain	6
4.3	GenericAPI	7
4.3.1	ServiceModuleHandler	7
4.3.2	ResourceController	7
4.3.3	LibVirtController	7
4.3.4	LibVirt<category>Controller	7
4.4	persistence	8
5	Dependency Injection	8
6	Deployment	9
7	Persistierung	9
8	Sequenzdiagramme	10
8.1	service bestellen	10

1.1 Zweck

Dieses Dokument beschreibt die Software Architektur für das Projekt SDDC.

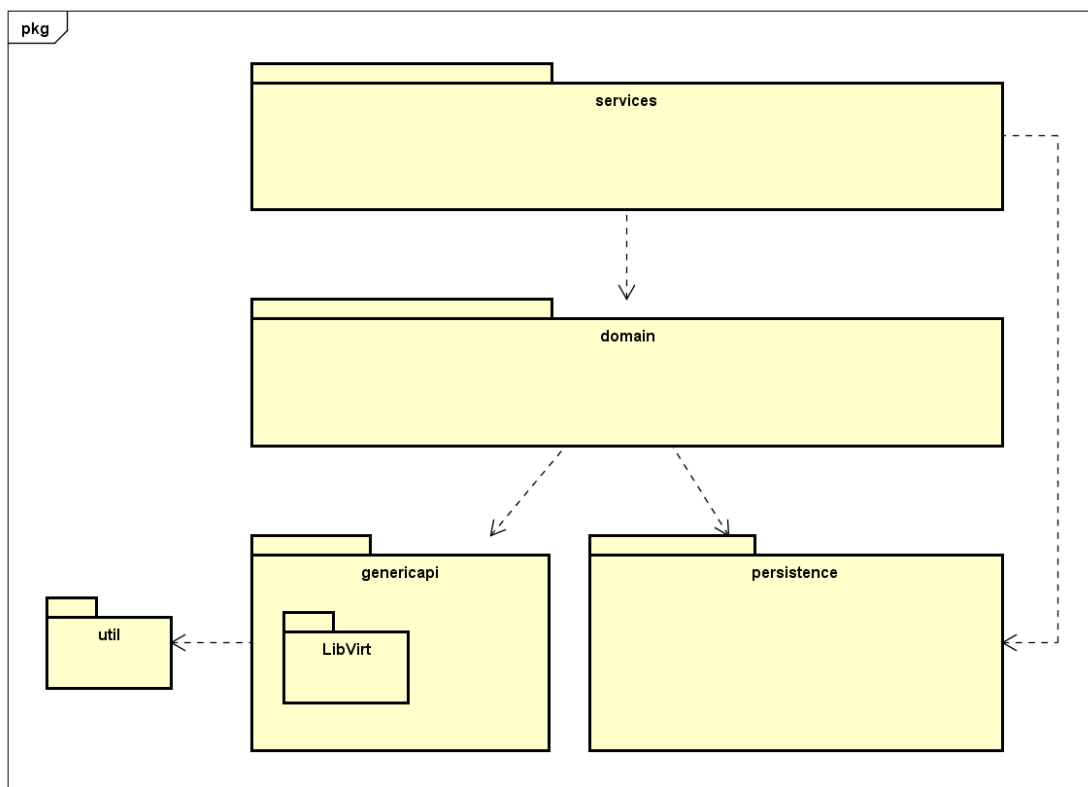
1.2 Gültigkeitsbereich

Dieses Dokument ist während des ganzen Projekts gültig und wird laufend aktualisiert.

2 Systemübersicht

3 Logische Architektur

- services: Web-Schnittstelle (RESTful)
- genericapi: Abstraktion für Compute, Storage und Network
- common: DTOs (Service, OrderedService, ServiceModule)
- util: Tools zur Bearbeitung von Configfiles
- presistence: OR-Mapping

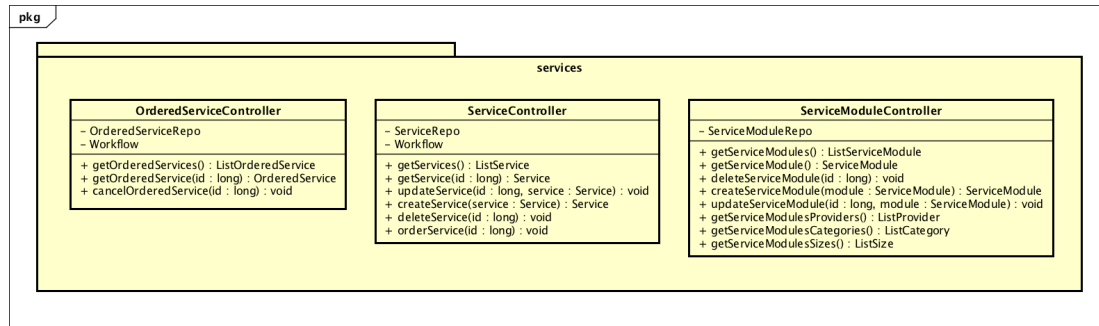


powered by Astah

4 Klassenstruktur

4.1 services

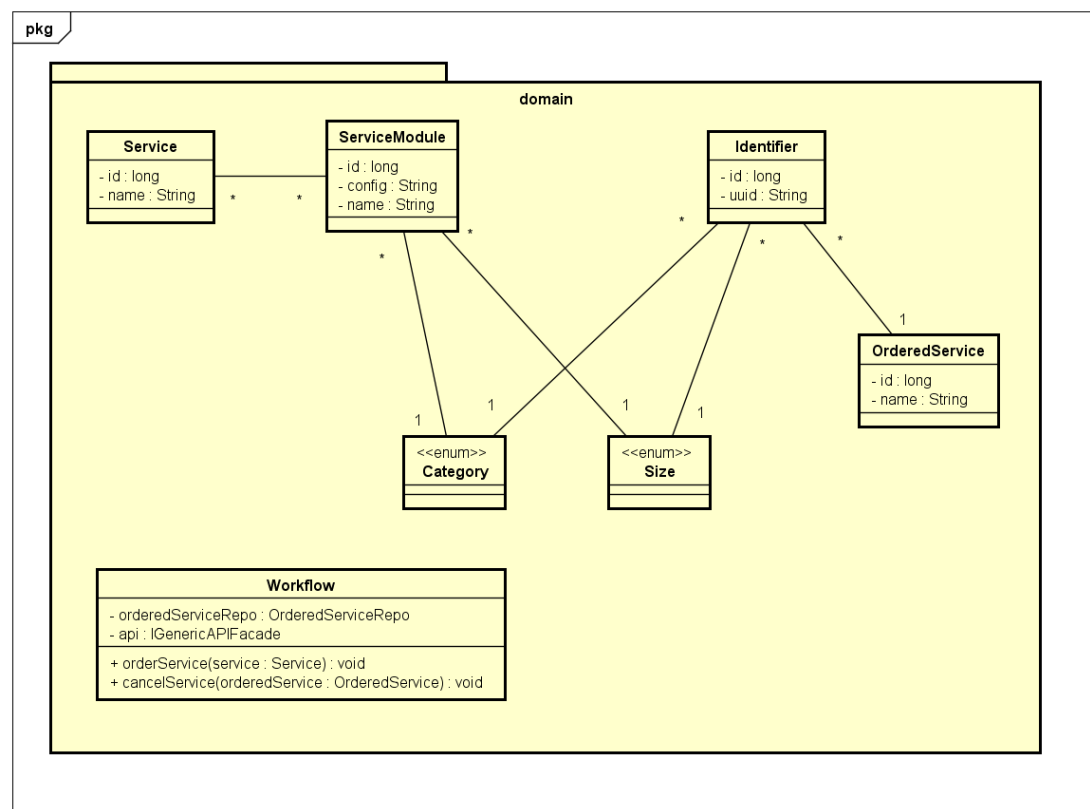
In der Schicht services wird die RestAPI implementiert.



powered by Astah

4.2 domain

Die Domain beinhaltet die Entitäten, die mittels OR-Mapper mit der Datenbank synchronisiert werden. Der Workflow erstellt/löscht einen Service.



powered by Astah

4.3 GenericAPI

4.3.1 ServiceModuleHandler

Der ServiceModuleHandler entscheidet an welchen ResourceController ein ServiceModule/Identifier übergeben wird. Der Entscheidungsalgorithmus kann auf verschiedenste Weise implementiert werden, deshalb gibt es ein Interface.

4.3.2 ResourceController

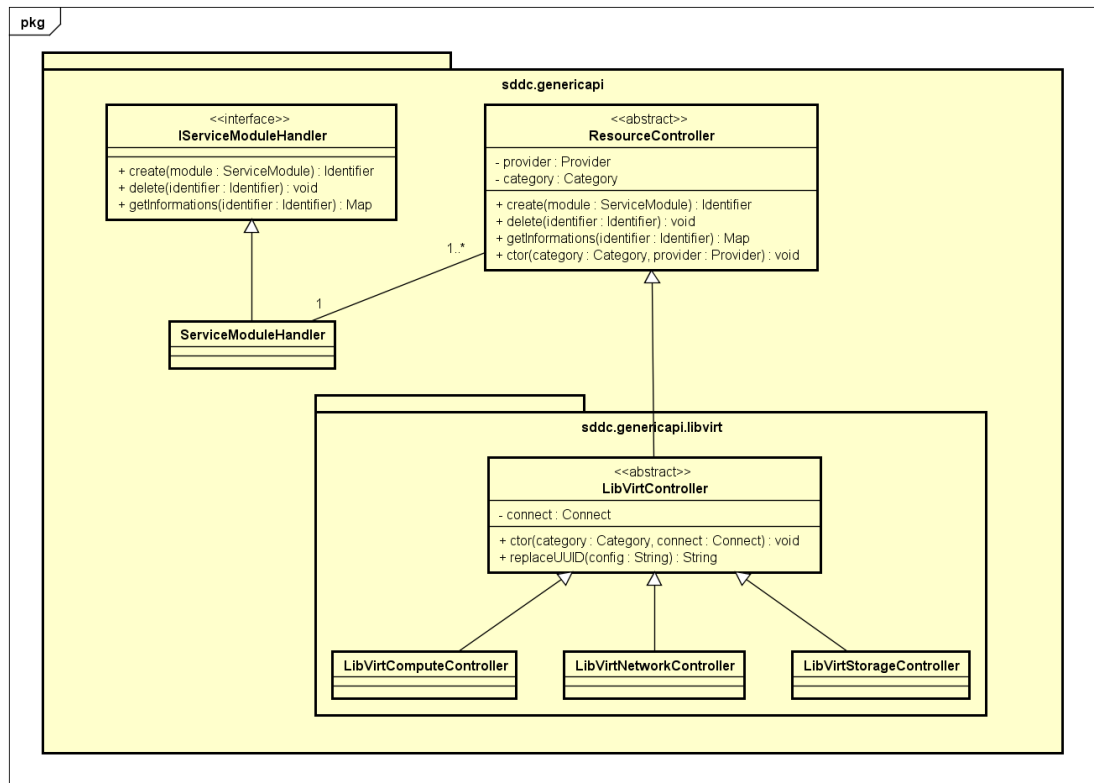
Der ResourceController ist eine Abstraktion eines Controllers, die nicht auf irgendeine Bibliotheksabhängigkeiten (z.B. LibVirt, JClouds) eingeht und lediglich dem ServiceModuleHandler die nötigen Informationen anbietet.

4.3.3 LibVirtController

Der LibVirtController ist eine weitere Abstraktion eines Controllers, die aber speziell einer Bibliothek (in diesem Fall LibVirt) zugewiesen ist. Die Klasse kümmert sich um die Instanziierung und kann, wenn nötig, Helfermethoden anbieten.

4.3.4 LibVirt<category>Controller

Die konkrete Klasse des Controllers implementiert nur noch die abstrakten Methoden des ResourceControllers.



powered by Astah

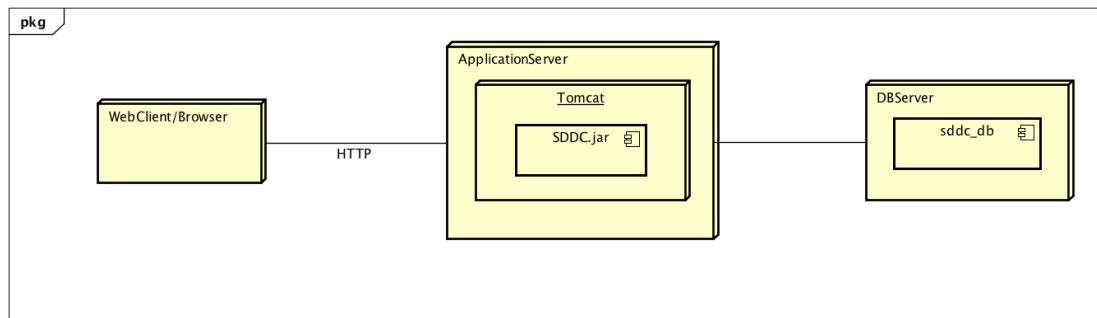
4.4 persistence

Wird durch OR Mapper (Hibernate) vorgenommen, dieser mappt die Java Objekte auf die Datenbank.

5 Dependency Injection

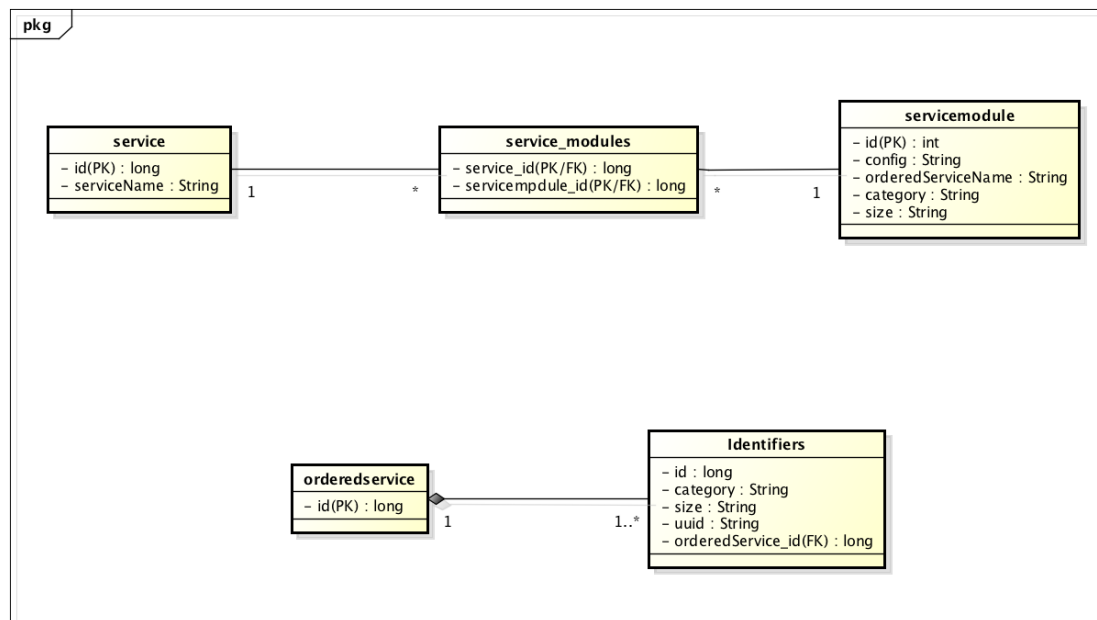
Die GenericAPI ist relativ komplex ausgefallen und kann erweitert werden. Um unnötigen Code in den oberen Schichten zu vermeiden, wird die GenericAPI mit Hilfe von Dependency Injection aufgesetzt. Es ist möglich in einem XML die gesamte GenericAPI nach belieben zusammenzusetzen.

6 Deployment



powered by Astah

7 Persistierung



powered by Astah

8 Sequenzdiagramme

8.1 service bestellen

