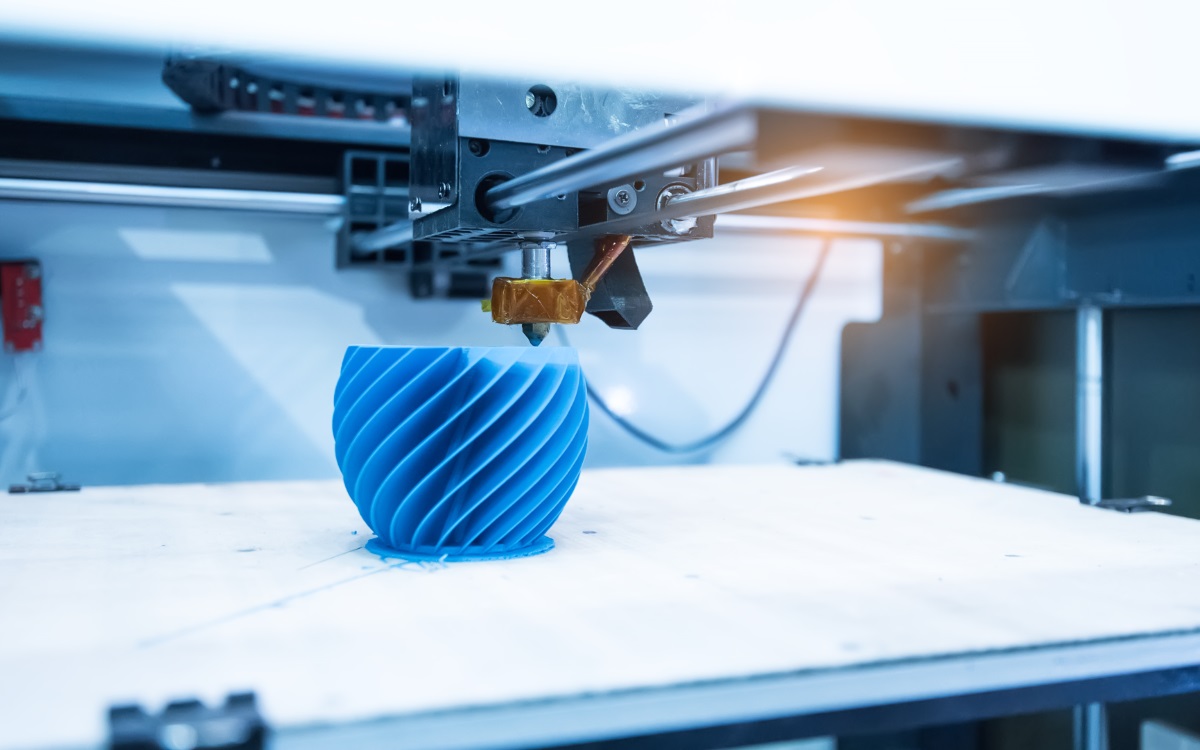
Cloud Computing – Teil 2

- Prof. Dr. Kasche –

**Dokumentation**

Thema: Skript zum outsourcen eines Slicing-Prozess (3D)



Kevin Lempert

Jonas Renz

- PIA18 -

Inhaltsverzeichnis

[1 Kurzinfo 3](#_Toc53490500)

[2 Bedienungsanleitung 4](#_Toc53490501)

[2.1 Anleitung 4](#_Toc53490502)

[2.2 Parameter 4](#_Toc53490503)

[2.3 Funktionsweise 4](#_Toc53490504)

[3 Projektdokumentation 5](#_Toc53490505)

[3.1 User Stories / Story-Board 5](#_Toc53490506)

[3.2 Kann-Board 5](#_Toc53490507)

[3.3 MVP 5](#_Toc53490508)

[4 Systemdokumentation 6](#_Toc53490509)

[4.1 Code / Code-Kommentare 6](#_Toc53490510)

[4.2 Wizard 6](#_Toc53490511)

[4.3 Architektur / PAP 6](#_Toc53490512)

[5 Schnittstellendokumentation 7](#_Toc53490513)

[5.1 AWS 7](#_Toc53490514)

[5.2 Python SDK 7](#_Toc53490515)

[6 Testdokumentation 8](#_Toc53490516)

1. Kurzinfo

|  |  |  |
| --- | --- | --- |
| **Autor:** |  | **Kevin Lempert Jonas G. Renz** |
| **Hochschuleinrichtung:** |  | **Duale Hochschule Gera-Eisenach (Campus Gera)** |
| **Version:** |  | **0.1** |
| **Programmiersprache:** |  | **Python 3.x** |
| **Datum:** |  | **26.10.2020** |
| **Systemaufruf:** |  |  |
| **Parameter (wichtigste):** |  | **https://www.styrz.de/grundlagen-der-softwaredokumentation/** |
|  |  |  |

1. Bedienungsanleitung
   1. Anleitung
   2. Parameter

|  |  |  |
| --- | --- | --- |
| Parameter | Variable(n) | Erklärung |
| --bed-temperature <t> | t = ganze Zahl, z.B.: 98 | Hier soll die Druckbett-Temperatur angegeben werden. Maßeinheit: [°C] |
| --cooling | - | Lüfter- und Kühlungssteuerung aktivieren |
| --filament-diameter <d> | d = ganze Zahl, z.B.: 6 | Durchmesser des Filaments  Maßeinheit: [mm] Default: 3 |
| --fill-density <d> | d = ganze Zahl, z.B.: 50 | Dichte der Füllung Maßeinheit: [%] Default: 40 Bereich: 0-100 |
| --layer-height <h> | h = Kommazahl, z.B.: 0.2 | Hier wird die Layer-Höhe des Modells angegeben. Maßeinheit: [mm] Default: 0.4 |
| --nozzle-diameter <d> | d = Kommazahl, z.B.: 0.35 | Durchmesser der Düse Maßeinheit: [mm] Default: 0.5 |
| --output <o> | o = Name der Datei | Datei, in die gcode ausgegeben werden soll Wenn ein Verzeichnis für diese Option angegeben wird, wird die Ausgabe unter diesem Verzeichnis gespeichert. |
| --output-filename-format <f> | f = Format der Datei  (gcode oder stl) | Hier wird das Format der berechneten Datei angegeben. |
| --retract-length <l> | l = ganze Zahl, z.B.: 36 | Länge des Rückzugs beim Anhalten der Extrusion Maßeinheit: [mm] |
| --scale-to-fit <x> <y> <z> | x/y/z = ganze Zahl, z.B.: 350 340 390 | Hier soll eine neue Größe für das Model angegeben werden.  Maßeinheit: [mm] |
| --support-material | - | Erzeugen von Hilfsmaterial für Überhänge |
| --temperature <t> | t = ganze Zahl, z.B.: 92 | Extrusionstemperatur  0 zum Deaktivieren  Maßeinheit: [°C] Default: 200 |

* 1. Funktionsweise

1. Projektdokumentation
   1. User Stories / Story-Board

Ein Bild, das drinnen, Screenshot, sitzend, Monitor enthält.

Automatisch generierte Beschreibung

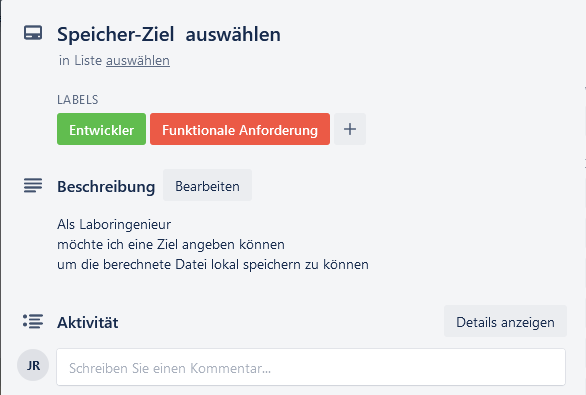
Das Story Board besteht aus einzelnen Karten, in dem der Kunde seine Anforderungen äußert.



Hier ist eine Karte aus dem Bereich „anzeigen“. Die Karte enthält eine nichtfunktionale Anforderung, die von den Entwicklern eingesetzt wurde. **Nicht-funktionale Anforderungen** hingegen sind meist unspezifisch für ein Produkt. Beispiele für nicht-funktionale Anforderungen sind User Stories enthalten immer den gleichen Text:

Als …  
möchte ich …  
um …

In der nächsten Karte ist eine funktionale Anforderung zu erkennen. **Funktionale Anforderungen** sind die Anforderungen, deren Umsetzung direkt der Zweckbestimmung des Produkts dienen. Sie sind spezifisch für dieses Produkt.



* 1. KanBan-Board

Ziel des Kanban-Boards ist es, Projektabläufe und Aufgaben zu visualisieren – in übersichtlichen Spalten mit einzelnen Einträgen, die der Reihe nach abgearbeitet werden können.

So sieht das Team auf einen Blick, was zu tun ist und in welchen Bereichen Probleme gelöst werden müssen. Dabei kann das Kanban Board auf viele verschiedene Weisen genutzt und gestaltet werden – um es optimal anzupassen und die Ergebnisse zu verbessern.

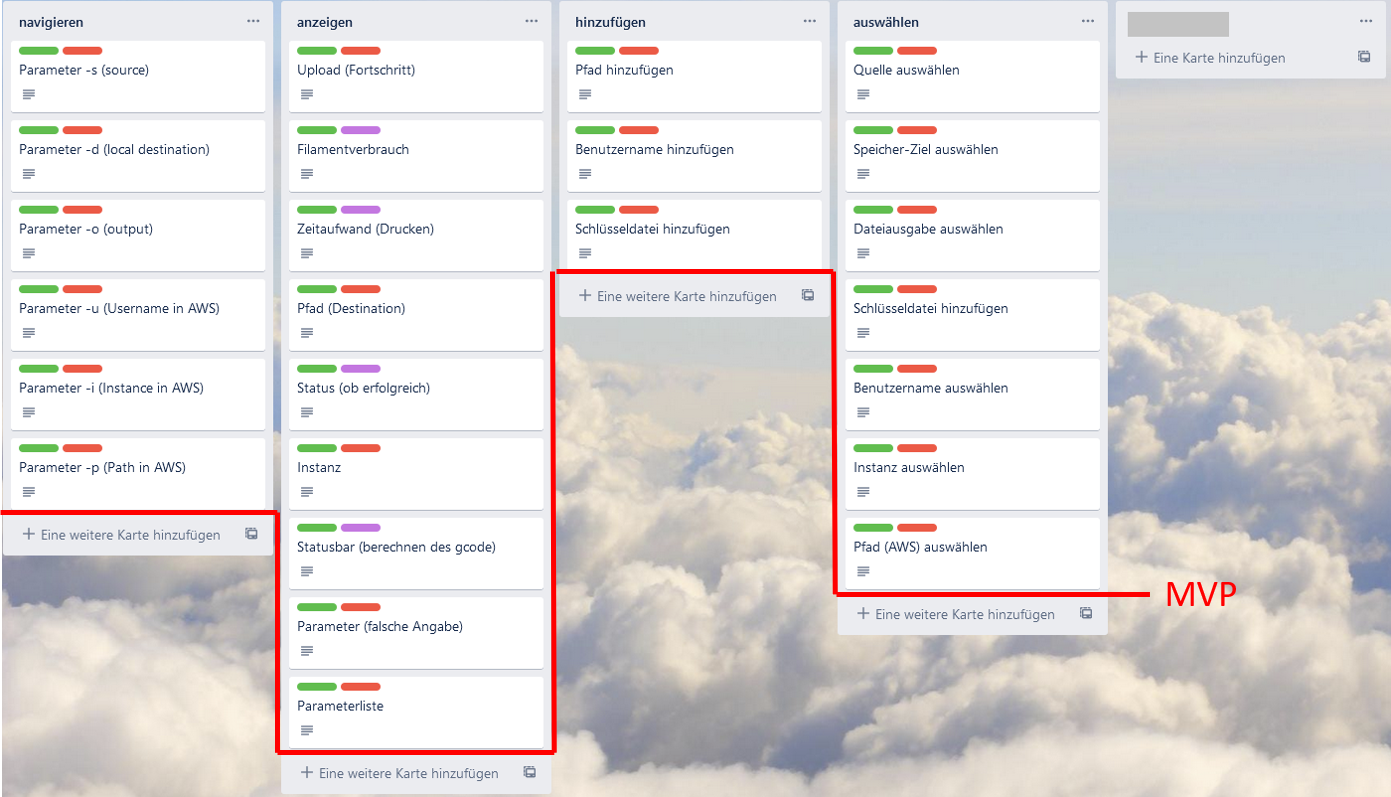
Hier ist ein kleiner Ausschnitt aus dem KanBan-Board zu sehen. Zur Übersichtlichkeit wurde aus der frühen Projektphase ein Foto übernommen.

Ein Bild, das Screenshot, drinnen, Monitor, Computer enthält.

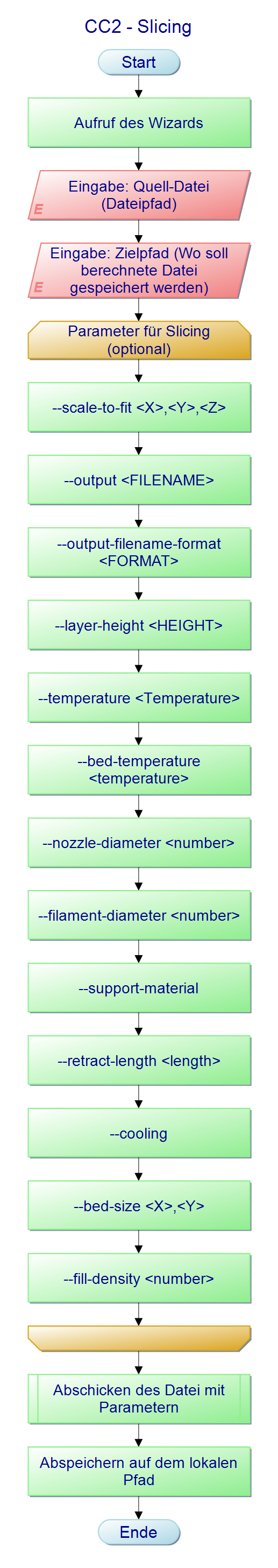
Automatisch generierte Beschreibung

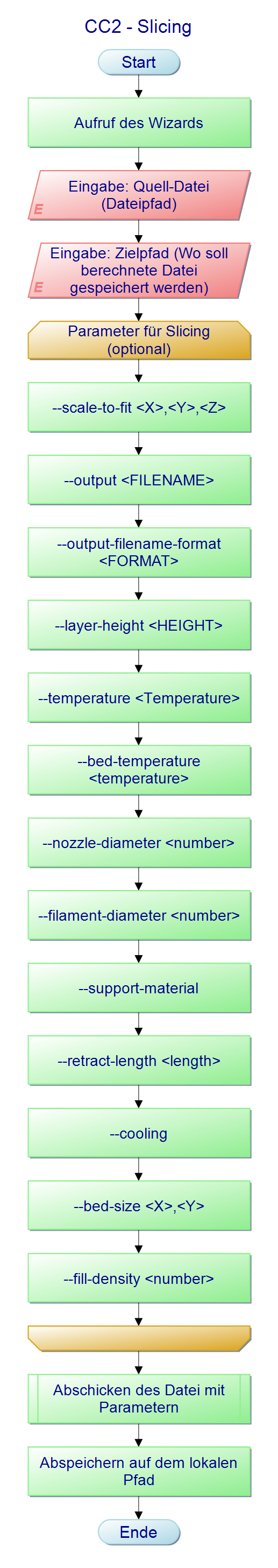
* 1. MVP

Ein Minimum Viable Product (MVP) ist die erste minimal funktionsfähige Iteration eines [Produkts](https://de.wikipedia.org/wiki/Produkt_(Wirtschaft)), das entwickelt werden muss, um mit minimalem Aufwand den Kunden-, Markt- oder Funktionsbedarf zu decken und handlungsrelevantes Feedback zu gewährleisten. Das MVP sieht ist aus dem Story-Board zu entnehmen.



1. Systemdokumentation
   1. Code / Code-Kommentare
   2. Wizard
   3. PAP





1. Schnittstellendokumentation
   1. AWS

**Amazon Web Services** (**AWS**) ist ein US-amerikanischer Cloud-Computing-Anbieter, der 2006 als Tochterunternehmen des Online-Versandhändlers Amazon.com gegründet wurde. Zahlreiche populäre Dienste wie beispielsweise Dropbox, Netflix, Foursquare oder Reddit greifen auf die Dienste von Amazon Web Services zurück. 2017 stufte Gartner AWS als führenden internationalen Anbieter im Cloud Computing ein.

Der Web-Service Amazon Elastic Compute Cloud (Amazon EC2) stellt sichere, skalierbare Rechenkapazitäten in der Cloud bereit. Der Service ist darauf ausgelegt, Cloud Computing für Entwickler zu erleichtern. Amazon EC2 bietet die breiteste und tiefste Computerplattform mit einer Auswahl an Prozessor, Speicher, Netzwerk, Betriebssystem und Kaufmodell.

* 1. Python SDK

Amazon stellt für Python-Skripte ein SDK zur Verfügung. Das SDK nennt sich Boto3. Boto3 erleichtert die Integration von Python-Anwendung, ‑Bibliothek oder Scripts in AWS-Services wie Amazon S3, Amazon EC2, Amazon DynamoDB und andere.

Boto3 hat zwei getrennte Ebenen von APIs. Die Client-APIs (oder API auf niedriger Ebene) bieten eine 1-zu-1-Zuordnung der zugrunde liegenden HTTP API-Vorgänge. Ressourcen-APIs verbergen explizite Netzwerkaufrufe und stellen stattdessen Ressourcenobjekte und Collections bereit, um auf Attribute zuzugreifen und Aktionen durchzuführen.

1. Testdokumentation