

# Automatisierung eines 3D-Druck-Labors

Matthias Emde



TUM



**Bachelorarbeit**

# **Automatisierung eines 3D-Druck-Labors**

Matthias Emde

18. Oktober 2019



Lehrstuhl für Datenverarbeitung  
Technische Universität München



Matthias Emde. *Automatisierung eines 3D-Druck-Labors*. Bachelorarbeit, Technische Universität München, München, 2019.

Betreut von Prof. Dr.-Ing. K. Diepold und Matthias Kissel / Stefan Röhl; eingereicht am 18. Oktober 2019 bei der Fakultät für Elektrotechnik und Informationstechnik der Technischen Universität München.

© 2019 Matthias Emde

Lehrstuhl für Datenverarbeitung, Technische Universität München, 80290 München, <http://www.ldv.ei.tum.de/>.

Dieses Werk ist unter einem Creative Commons Namensnennung 3.0 Deutschland Lizenzvertrag lizenziert. Um die Lizenz anzusehen, gehen Sie bitte zu <http://creativecommons.org/licenses/by/3.0/de/> oder schicken Sie einen Brief an Creative Commons, 171 Second Street, Suite 300, San Francisco, California 94105, USA.

# Zusammenfassung

Die stetig wachsende Relevanz der 3D-Druck-Technologie für die Forschung und Industrie erfordert von der Lehre, Studenten mit den Möglichkeiten des 3D-Drucks vertraut zu machen. Zu diesem Zweck wurde vom *Lehrstuhl für Datenverarbeitung* ein 3D-Druck-Labor aufgebaut. Dem Labor fehlt jedoch eine Infrastruktur zur effizienten Bedienung, Überwachung und Verwaltung der Drucker. In dieser Arbeit wird deswegen ein geeignetes Verfahren zur Automatisierung des 3D-Druck-Labors ausgewählt und in Form eines Prototypen umgesetzt.

Die Automatisierung erfolgt durch eine im Rahmen dieser Arbeit entstandenen Webanwendung. Diese steuert und kontrolliert den Informationsfluss zwischen den Studenten und den 3D-Druckern. Die Benutzeroberfläche der Anwendung stellt alle Funktionen für die Steuerung des 3D-Druck-Prozesses sowie eine Drucker- und Benutzerverwaltung zur Verfügung.

Die in dieser Arbeit erfolgreich durchgeführte Umsetzung des Prototyps zur Automatisierung des 3D-Druck-Labors erschließt viele Anwendungsmöglichkeiten. Durch Benutzerprofile kann für die Benutzer vorgegeben werden, welche Drucker sie verwenden und welche Einstellungen sie während des 3D-Druck-Prozesses vornehmen dürfen. Durch Protokollierung und Auswertung der Aktivitäten im Labor lassen sich Erkenntnisse, über das Lernverhalten der Studenten gewinnen, welche wiederum in die Gestaltung der Benutzerprofile einfließen können.



# Inhaltsverzeichnis

<b>Kurzzusammenfassung</b>	<b>3</b>
<b>1 Einführung</b>	<b>7</b>
1.1 Motivation . . . . .	7
1.2 Stand der Technik . . . . .	8
<b>2 3D-Druck-Prozess</b>	<b>11</b>
<b>3 Grundlagen für die Automatisierung</b>	<b>15</b>
3.1 Druckerserver . . . . .	15
3.2 Webanwendung . . . . .	18
3.2.1 Backend . . . . .	19
3.2.2 Frontend . . . . .	19
3.3 Kommunikation und Sicherheit . . . . .	21
<b>4 Funktionsweise der Webanwendung</b>	<b>23</b>
4.1 Kommunikation zwischen Web- und Druckerserver . . . . .	23
4.1.1 Programmierschnittstelle . . . . .	23
4.1.2 Websocket . . . . .	24
4.2 Automatisierung des 3D-Druck-Prozesses . . . . .	25
4.2.1 Slicing-Prozess . . . . .	25
4.2.2 Start des Drucks . . . . .	28
4.3 Kommunikation zwischen Webserver und Benutzeroberfläche . . . . .	28
4.3.1 Programmierschnittstelle . . . . .	29
4.3.2 Benutzerverwaltung . . . . .	29
4.3.3 Websocket . . . . .	30
4.4 Frontend . . . . .	31
4.4.1 Basis-Layout . . . . .	31
4.4.2 Komponenten . . . . .	32
<b>5 Anwendung im 3D-Druck-Labor</b>	<b>39</b>
5.1 Benutzerprofile . . . . .	39
5.2 Protokollierung und Auswertung . . . . .	40
5.3 Kontrolle der Drucker . . . . .	40

*Inhaltsverzeichnis*

<b>6</b>	<b>Fazit und Erweiterungen für den Betrieb</b>	<b>41</b>
6.1	Datenbank . . . . .	41
6.2	Authentifikation . . . . .	42
6.3	Anpassung des Layouts an mobile Endgeräte . . . . .	42
6.4	Druckkostenberechnung . . . . .	43



# 1 Einführung

## 1.1 Motivation

Die 3D-Druck-Technologie hat sich in den letzten Jahren stark weiterentwickelt und ist mittlerweile sowohl in der Forschung als auch in der Industrie unerlässlich. Durch den vergleichsweise schnellen Fertigungsprozess ist ein 3D-Drucker das ideale Werkzeug, um Ideen zu testen (Rapid Prototyping) oder ganze Projekte umzusetzen. Deshalb wird es im akademischen Umfeld als wichtig erachtet, Studenten der Ingenieurwissenschaften mit dem Umgang und den Möglichkeiten der 3D-Druck-Technologie vertraut zu machen.

Die am *Lehrstuhl für Datenverarbeitung* durchgeführten 3D-Druck-Workshops bieten den Studenten bereits einen guten Einstieg in das Feld. Um das Gelernte zu vertiefen und sich mit allen Aspekten der Arbeit mit 3D-Druckern vertraut zu machen, ist es für die Studenten allerdings zwingend notwendig, eigenständig mit 3D-Druckern zu arbeiten und dabei eigene Erfahrungen zu sammeln.

In Verfolgung dieses Ziels wurde am *Lehrstuhl für Datenverarbeitung* ein 3D-Druck-Labor aufgebaut (siehe Abb. 1.1). Für die Studenten stehen hier zur Zeit mehr als zehn 3D-Drucker bereit. Die Bedienung der Maschinen erfordert ohne eine ergänzende Infrastruktur viele manuell auszuführende Arbeitsschritte, die zum einen für die Studenten eine Hemmschwelle darstellen und sich zum anderen nicht ohne die Anwesenheit eines Betreuers kontrollieren lassen.

Deswegen wird eine zentrale Instanz benötigt, durch die alle Aktivitäten im 3D-Druck-Labor gesteuert und durch den Lehrstuhl kontrolliert werden können. Diese Steuereinheit sollte sich idealerweise von überall erreichen lassen, da es gerade bei 3D-Drucken, die viele Stunden dauern können, äußerst nützlich ist, den Druckverlauf auch von außerhalb des Labors verfolgen zu können.

In dieser Arbeit sollen nun die Möglichkeiten zur Realisierung einer solchen Instanz und damit der Automatisierung des 3D-Druck-Labors untersucht werden. Der am besten geeignete Ansatz soll dann als Prototyp umgesetzt werden, um die Funktionalität testen zu können.

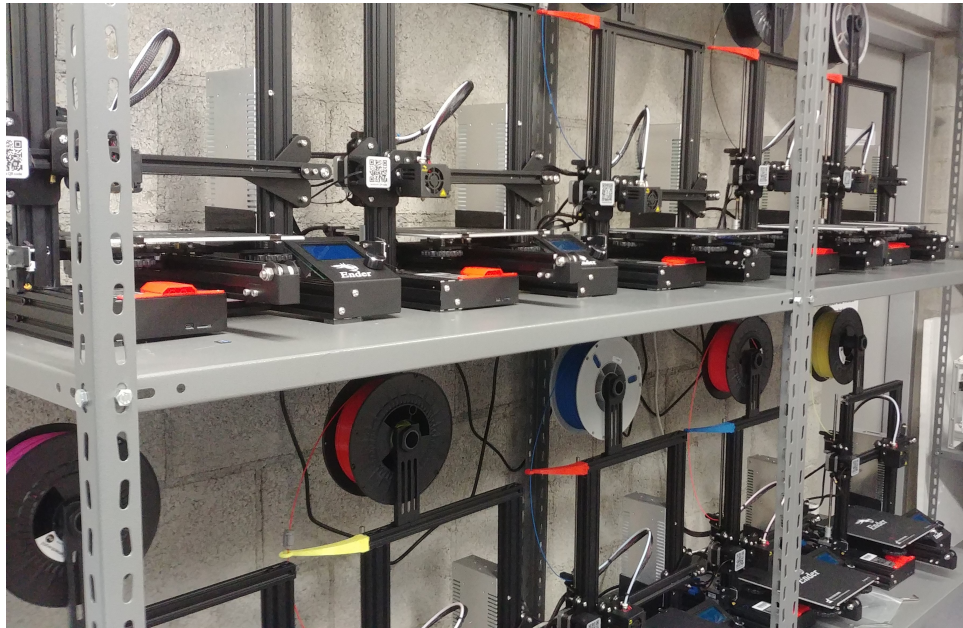


Abbildung 1.1: Das 3D-Druck-Labor des Lehrstuhls für Datenverarbeitung

## 1.2 Stand der Technik

Möchte man ein Werkstück mit einer bestimmten, exakten Form maschinell produzieren, gibt es dafür grundsätzlich zwei Verfahren:

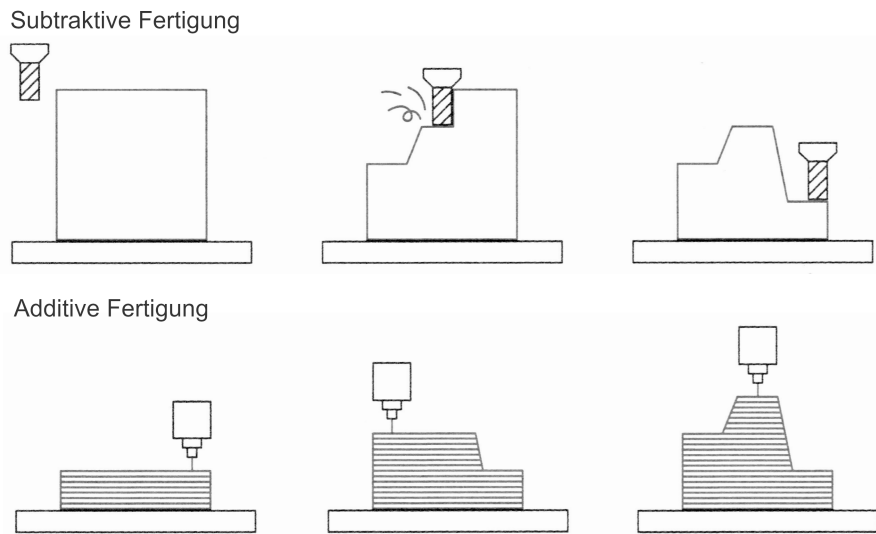
Beim ersten Verfahren beginnt man mit einem festen Rohmaterial, von dem z.B. mit einer Fräse nach und nach alles entfernt wird, was nicht zum gewünschten Objekt gehört. Diese Vorgehensweise bezeichnet man daher auch als subtraktive Fertigung.

Das zweite Verfahren besteht darin, das Werkstück Schritt für Schritt durch Hinzufügen von Material aufzubauen (additive Fertigung). Hierbei muss das Rohmaterial in einen formbaren Zustand gebracht werden und nach der Platzierung z.B. durch Abkühlung aushärten. Ein schematischer Vergleich der beiden Verfahren ist in Abbildung 1.2 zu sehen.<sup>1</sup>

Die zu erreichende Komplexität der Werkstücke ist bei additiven Fertigungsver-

---

<sup>1</sup>Der Vollständigkeit halber müsste in dieser Aufzählung auch noch die formative Fertigung aufgeführt werden, bei der das flüssige Rohmaterial in eine Form gegossen wird und darin aushärtet. Da eine solche Form aber wiederum durch maschinelle Fertigung hergestellt werden muss, wird die formative Fertigung hier nicht weiter betrachtet.



**Abbildung 1.2:** Vergleich der beiden Fertigungsverfahren

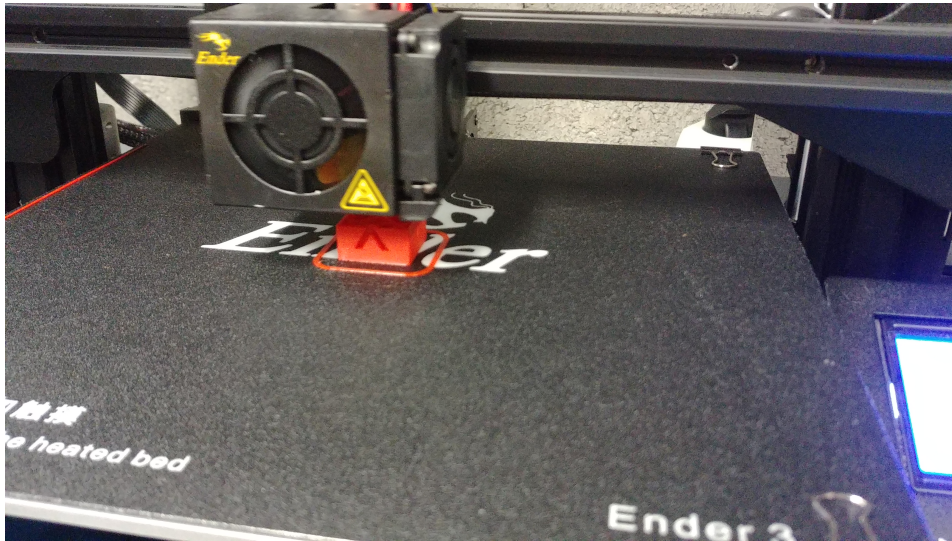
fahren deutlich höher als bei subtraktiven, da im subtraktiven Prozess z.B. das Innere von geschlossenen Körpern durch das Werkzeug nicht erreichbar sind. Die additive Fertigung verbraucht außerdem weniger Rohmaterial, da nichts entfernt werden muss.

Ein 3D-Drucker ist eine Maschine, die einen additiven Fertigungsprozess durchführt. 3D-Drucker untergliedern sich weiter in Typen, die verschiedene Materialien auf unterschiedliche Weise aufschichten. In dieser Arbeit werden Drucker vom Typ *Fused Deposition Modelling (FDM)* betrachtet; dieser Typ wird aufgrund einer Patentstreitigkeit<sup>2</sup> auch *Fused Filament Fabrication (FFF)* genannt. Auch unter den FDM/FFF-Druckern gibt es viele verschiedene Bauweisen (Delta, XYZ, etc..), wobei in dieser Arbeit ausschließlich die am weitest verbreiteten XYZ-FDM-Drucker beschrieben werden. Im Folgenden wird hierfür nur noch der allgemeine Begriff *3D-Drucker* verwendet.

3D-Drucker benutzen einen ca. 200 °C heißen Druckkopf (engl. Hotend), um ein auf Kunststoff basierendes Rohmaterial (Filament) zu schmelzen und hinauszudrücken (extrudieren). Das Filament wird von einem als Extruder bezeichneten Motor mit Hilfe eines Zahnrads in das Hotend geschoben. Die Position des Druckkopfs relativ zum Druckbett wird dabei in den drei Dimensionen X, Y und Z mit jeweils einem Schrittmotor präzise kontrolliert.

<sup>2</sup><https://3druck.com/blog/unterschied-fff-fdm-4183138/>

## 1 Einführung



**Abbildung 1.3:** Ein 3D-Drucker im Druckprozess

Im Druckprozess wird das Filament zunächst in der XY-Ebene in Bahnen auf das Druckbett gelegt, bis die erste Schicht (engl. Layer) vollendet ist. Danach wird je nach Drucker-Modell entweder der Druckkopf nach oben oder das Druckbett nach unten bewegt (Z-Achse) und das nächste Layer auf das jeweils vorherige gedruckt.

Damit das Druck-Objekt gut auf dem Druckbett haftet, wird das Druckbett beheizt. Die Temperatur hierfür variiert mit dem verwendeten Kunststoff, liegt aber generell in einem Bereich zwischen 50 °C und 100 °C. Die beheizte Druckplattform (engl. Heatbed) verhindert außerdem, dass das Druck-Objekt zu schnell abkühlt, sich dabei verformt und vom Heatbed ablöst.

Andererseits muss das aus dem Hotend austretende flüssige Filament durch einen Lüfter aktiv gekühlt werden, damit es schnell aushärtet und die vorgegebene Form beibehält.

## 2 3D-Druck-Prozess

Nachdem die grundsätzliche Funktionsweise eines 3D-Druckers im vorherigen Kapitel erklärt wurde, kann der Ablauf des gesamten Druck-Prozesses von der Auswahl des Objekts bis zu dessen Fertigstellung beschrieben werden.

Ein 3D-Objekt kann entweder selbst in einem *Computer-aided design (CAD)*-Programm entworfen oder von einer Online-Plattform wie *Thingiverse*<sup>1</sup> heruntergeladen werden. Das gebräuchliche Datei-Format hierfür ist STL - kurz für Stereolithographie.

Die STL-Datei wird mit Hilfe eines Slicer-Programms in Schichten geschnitten (engl. slicing) und in vom 3D-Drucker lesbaren Maschinencode (Gcode) umgewandelt. Der Gcode wird in der Slicer-Software als dreidimensionale Druckvorschau angezeigt (siehe Abb. 2.1).

Der Gcode besteht aus einem Datei-Header, in dem z.B. die Temperaturen für das Hotend und Heatbed gesetzt werden. Darunter folgt der Body, in dem die Bewegungs-Anweisungen für die vier Motoren stehen (X-,Y- und Z-Achse, Extruder (E)), die das Objekt erzeugen. Im folgenden Beispiel sind Ausschnitte für die beiden Datei-Komponenten zu sehen. Zeilen mit einem Semikolon sind Kommentare und werden vom 3D-Drucker ignoriert.

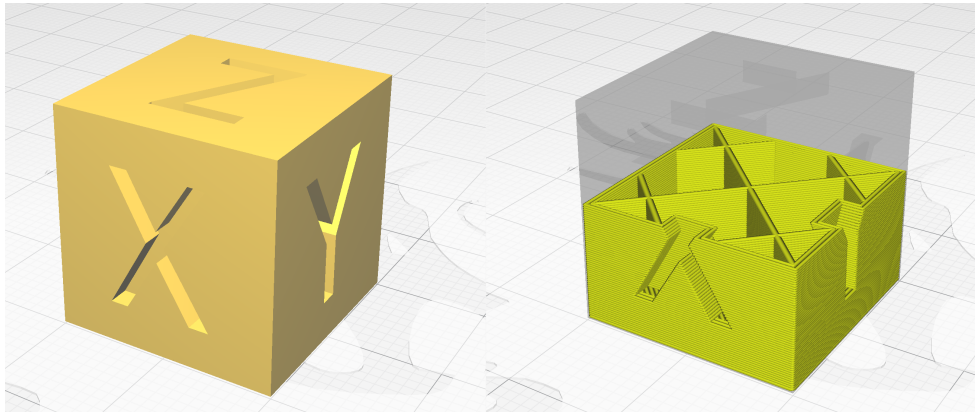
```
; Header
M104 S204
M109 S204
G28
...

; Body
G1 Z1.380 F1002
G1 X161.361 Y132.925 E3.7788 F450
G1 X162.728 Y133.918 E4.0598
...
```

---

<sup>1</sup><https://thingiverse.com>

## 2 3D-Druck-Prozess



**Abbildung 2.1:** Die STL-Datei eines Würfels und der aus dem Slicing resultierende Gcode, geladen in der Slicer-Software *Cura*

Im Header-Ausschnitt wird das Kommando M104 verwendet, um die Temperatur des Druckkopfs auf den Wert 204 zu setzen. Der Befehl M109 weist den Drucker an zu warten, bis die im S-Parameter übergebene Temperatur erreicht ist, um dann mit G28 die bewegten Einheiten für alle Achsen auf ihre Start-Position zu bringen.

Der Body enthält fast ausschließlich G1-Befehle, die den Drucker anweisen, eine lineare Bewegung auszuführen. Die übergebenen Parameter sind die Zielpositionen der vier Motoren (X, Y, Z und E) und die Geschwindigkeit, mit der die Bewegung ausgeführt wird (F).<sup>2</sup>

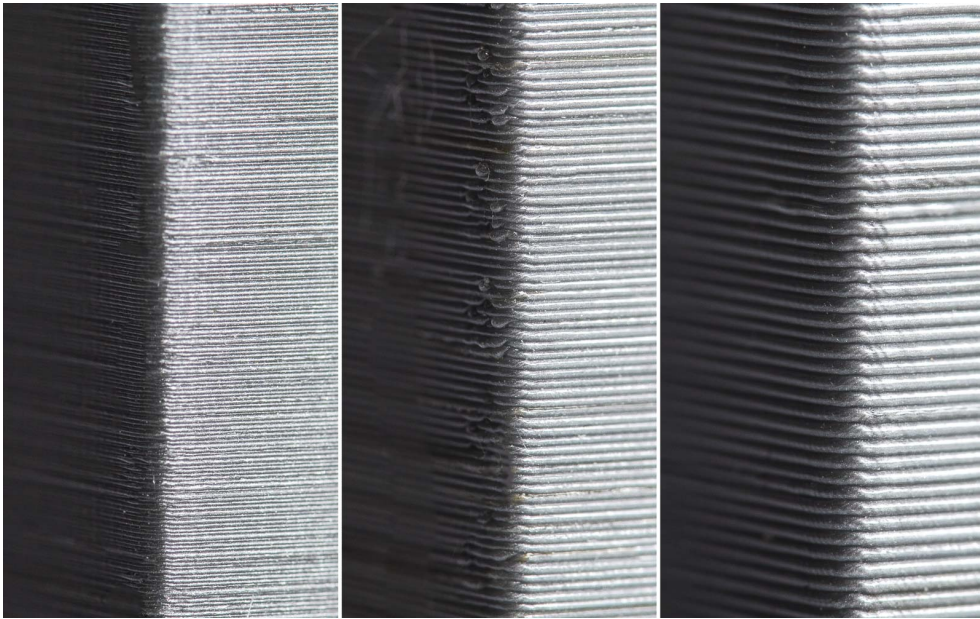
Für die Erzeugung des Gcodes stellt der Slicer ein großes Maß an einstellbaren Parametern bereit, mit denen sich die Qualität und die allgemeinen mechanischen Eigenschaften des Druckergebnisses stark beeinflussen lassen. Um erfolgreich mit 3D-Druckern arbeiten zu können, ist ein tieferes Verständnis der Slicer-Einstellungen und deren Auswirkungen unerlässlich.

Um eine Vorstellung von den Einstellungsmöglichkeiten zu vermitteln, werden im Folgenden einige Parameter vorgestellt:

**Die Schichthöhe (engl. Layer Height)** legt die Dicke der einzelnen Layer fest und variiert üblicherweise zwischen 0,1 und 0,3 mm. Je feiner die Schichten, desto glatter sind die Oberflächen und desto deutlicher sind Details erkennbar. Die Unterschiede in den Oberflächen sind in Abbildung 2.2 zu sehen. Natürlich müssen bei

---

<sup>2</sup>Eine vollständige Liste der Gcode-Befehle, die von der weit verbreiteten Drucker-Firmware *Marlin* akzeptiert werden, findet sich unter <https://marlinfw.org/meta/gcode/>



**Abbildung 2.2:** Unterschiede in der Oberfläche durch verschiedene Schichthöhen

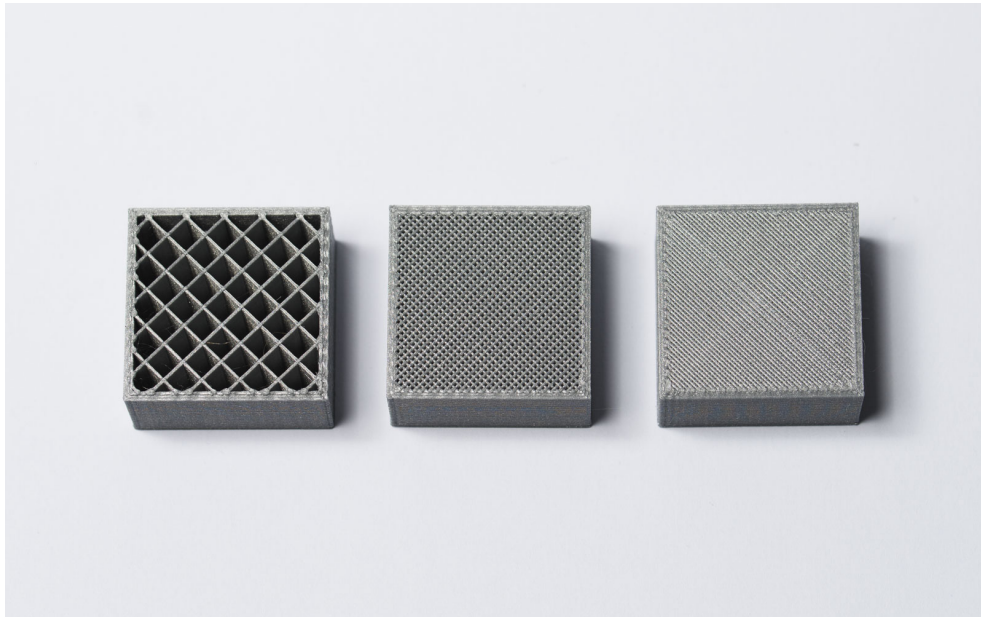
einer kleineren Schichthöhe auch deutlich mehr Schichten gedruckt werden, was entsprechend mehr Zeit in Anspruch nimmt.

**Die Wanddicke (engl. Shell Thickness):** Um Zeit, Kosten und Gewicht zu sparen, werden 3D-Objekte grundsätzlich nicht massiv gedruckt. Stattdessen werden Außenwände und eine Füllstruktur im Inneren gedruckt. Die Shell Thickness beschreibt die Dicke dieser Wände bzw. die Anzahl der Bahnen (Perimeter), die als Außenwand gelegt werden. Die Anzahl der Perimeter hat großen Einfluss auf die Stabilität des fertigen Objekts. In der Regel erzielen zwei bis drei Bahnen ein gutes Ergebnis, was bei einer Schicht-Breite von 0,4 mm eine Wanddicke zwischen 0,8 und 1,2 mm ergibt.

**Der Füllgrad (engl. Infill-Percentage)** gibt an, zu wieviel Prozent das Objekt mit Filament ausgefüllt werden soll. Der Infill trägt nicht nur zur Stabilität bei, sondern gibt auch waagerechten Flächen auf der Oberseite des Objekts Halt, da diese sonst in die Luft gedruckt werden müssten. In Abbildung 2.3 ist das Innere eines Würfels mit verschiedenen Füllgraden zu sehen.

Üblicherweise gibt es zwei Methoden, dem 3D-Drucker den vom Slicer erzeugten Gcode zur Verfügung zu stellen. Entweder die Datei wird manuell auf eine SD-Karte

## 2 3D-Druck-Prozess



**Abbildung 2.3:** Verschiedene Füllgrade von 25% (links) über 50% (Mitte) bis 75% (rechts)

geladen, die dann in den Leser am Drucker geschoben und von dort durch den Drucker gelesen wird, oder der Drucker wird über USB mit einem Computer verbunden, welcher mit einer entsprechenden Software die Befehle aus dem Gcode an den Drucker sendet. Der restliche Prozess läuft automatisch auf dem Drucker ab. Umfassende Informationen zu 3D-Druckern und dem 3D-Druck-Prozess finden sich in [4] und [2].



## 3 Grundlagen für die Automatisierung

Automatisierung bedeutet hier, alle Arbeitsschritte des 3D-Druck-Prozesses in einer Webanwendung zusammenzufassen, über die sich auch eine große Zahl von 3D-Druckern von überall aus steuern lässt.

Als zentrale Einheit wird hierfür ein Webserver benötigt, der Druckaufträge und Steuerbefehle vom Benutzer annimmt und in der Lage ist, die 3D-Drucker zu kontrollieren. Obwohl es möglich wäre, alle Drucker per USB direkt mit dem Server zu verbinden und über diesen zu steuern, würde dies zu mehreren Problemen führen:

- Die Auslastung des Webservers wächst mit der Anzahl an aktiven Druckern, was die Skalierbarkeit des Systems limitiert
- Im Falle eines Server-Absturzes werden alle laufenden Druckaufträge abgebrochen

Um die Rechenzeit aufzuteilen und die Auswirkungen von Hardware-Fehlern lokal zu halten, wird jeder 3D-Drucker von einem eigenen Druckerserver gesteuert. Druckerserver und 3D-Drucker bilden zusammen eine autarke, an das Netzwerk angebundene Einheit.

Eine schematische Darstellung der eben beschriebenen Komponenten ist in Abbildung 3.1 zu sehen. Im folgenden Kapitel wird die Wahl der Hard- und Software für die einzelnen Komponenten und deren Vernetzung beschrieben.

### 3.1 Druckerserver

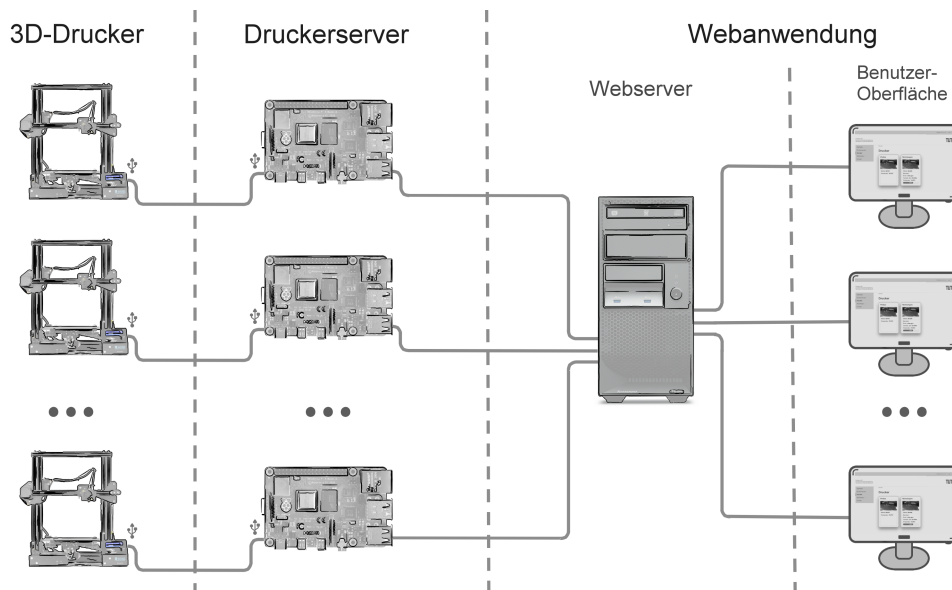
Um die automatische Steuerung der 3D-Drucker zu ermöglichen, werden diese über USB an einen Druckerserver angeschlossen. Da ein Druckerserver nur für jeweils einen Drucker zuständig ist, reicht hier schon die Rechenleistung aus, die ein Ein-Platinen-Computer wie der *Raspberry Pi*<sup>1</sup> zur Verfügung stellt.

Für die Druckersteuerungs-Software gibt es Anbieter aus dem kommerziellen

---

<sup>1</sup><https://raspberrypi.org>

### 3 Grundlagen für die Automatisierung



**Abbildung 3.1:** Übersicht der Hardwarekomponenten

und dem Open-Source-Bereich. Im Folgenden wird eine Auswahl vorgestellt und ihre Eignung für das 3D-Labor beurteilt.

**Astro-Print<sup>2</sup>** bietet bereits eine Weboberfläche zur Steuerung mehrerer 3D-Drucker. Allerdings ist diese für ein industrielles und nicht für ein akademisches Umfeld ausgelegt. So bietet sie insbesondere keine Möglichkeit, für einen einzelnen Benutzer zu definieren, welche Einstellungen er an einem Drucker vornehmen darf. Der Online-Service beruht auf einem Abo-System mit hohen monatlichen Kosten. Außerdem müssen für jeden Drucker eigene Druckerserver, sogenannte *Astro-Boxen*, erworben werden.

**Pro:**

- Ermöglicht zentrale Steuerung
- Erweiterungsmöglichkeiten durch Plugins

**Contra:**

- Eingeschränkte Personalisierbarkeit
- Sehr teuer, laufende Kosten

---

<sup>2</sup><https://astroprint.com>

**Repetier-Server**<sup>3</sup> ist eine Software, die mit ihren geringen Systemanforderungen auf einem Ein-Platinen-Computer laufen kann. Sie bietet ein übersichtliches Webinterface, ein für die Anbindung an eine eigene Software benötigte Programmierschnittstelle und die Möglichkeit, mehrere Drucker gleichzeitig über USB anzusteuern. Allerdings ist hier keine Slicer-Software enthalten, und für wichtige Funktionen wie den Webcam-Support wird die teure Premium-Version benötigt.

**Pro:**

- Übersichtliches Webinterface
- Programmierschnittstelle

**Contra:**

- Premium-Lizenz sehr teuer
- Noch kein Slicing-Support

**Octoprint**<sup>4</sup> ist eine Open-Source-Software mit einer aktiven Entwickler- und Benutzergemeinschaft. Die Software wird beständig weiterentwickelt und durch Plugins ergänzt. Sie besitzt ein Webinterface, auf dem ein Administrator komplette Kontrolle über den Drucker hat, aber auch eine einfach anzusprechende API, die für die Steuerung über den Webserver genutzt werden kann. *Octoprint* wird in Form des Betriebssystems *Octopi* speziell für den *Raspberry Pi* ausgeliefert. Es enthält eine Version des Slicer-Programms *Cura*<sup>5</sup>, mit dem 3D-Objekte direkt auf dem Druckerserver in Gcode umgewandelt (gesliced) werden können. *Octoprint* unterstützt zwar nur eine Drucker-Verbindung, doch aus Gründen der Stabilität und Modularität sollte generell nie mehr als ein Drucker pro Server angeschlossen werden.

**Pro:**

- Open-Source
- Umfangreiches API
- Integrierter Slicer

**Contra:**

- Nicht akutelle Version des Slicers

---

<sup>3</sup><https://repetier-server.com>

<sup>4</sup><https://octoprint.org>

<sup>5</sup><https://ultimaker.com/software/ultimaker-cura>

**Entscheidung:** Die Software *Octoprint* stellt alle benötigten Funktionen zur Verfügung, hat keine gravierenden Nachteile und ist kostenlos erhältlich. Deshalb wird sie bzw. das Betriebssystem *Octopi* in dem in dieser Arbeit vorgestellten Aufbau für die Druckerserver verwendet.

## 3.2 Webanwendung

Die Webanwendung bildet die zentrale Verbindungseinheit zwischen den Benutzern und den Druckerservern im 3D-Druck-Labor. Sie besteht wiederum aus zwei Komponenten. Die sichtbare Komponente ist die Benutzeroberfläche (engl. Frontend), die vom Browser dargestellt wird, sobald der Benutzer die Anwendung aufruft. Aufgaben wie die Nutzerverwaltung und die Kommunikation mit den Druckerservern werden von der anderen Komponente, dem sogenannten Backend, ausgeführt. Als Backend wird eine entsprechende Software bezeichnet, die auf dem Webserver ausgeführt wird. Der Webserver kann entweder auf einem lokalen Rechner oder bei einem Service-Anbieter im Internet gehostet werden.

Sowohl für den vom Server ausgeführten Code als auch für die Benutzeroberfläche gibt es die Option, für die Entwicklung ein Framework zu verwenden. Ein Framework ist eine Bibliothek, die in ihrem Anwendungsbereich benötigte Grundfunktionen bereitstellt und so den Programmieraufwand deutlich verringert. Vor- und Nachteile der Verwendung eines Frameworks werden im Folgenden aufgezeigt.

#### Pro:

- Auf einfache Verwendung ausgelegt
- Ein Leser mit Vorkenntnissen über das Framework kann den resultierenden Code leicht verstehen
- Verwendung der Framework-Funktionen bereits gut dokumentiert
- Beachtung wichtiger Sicherheitsaspekte für Webanwendungen ist bei einem Framework integriert
- Front-und Backend-Frameworks sind meist gut miteinander kompatibel

#### Contra:

- Initialaufwand für das Erlernen des Frameworks
- Eventuelle Probleme bei der Fehlersuche in uneinsehbaren Framework-Funktionen (Blackbox)

- Festlegung in der Vorgehensweise durch Verwendungen von vorgegebenen Funktionen

Insbesondere bei der Entwicklung einer Anwendung von geplanter Komplexität lohnt es sich, den Initialaufwand für das Erlernen und das Risiko eventueller Probleme in Kauf zu nehmen, um die vielen Vorteile eines Frameworks zu nutzen.

### 3.2.1 Backend

Für die Umsetzung der vom Webserver ausgeführten Software gibt es zwei Ansätze: *Python*<sup>6</sup> oder *JavaScript*<sup>7</sup>. Für beide Sprachen gibt es defacto Standard-Frameworks. *Django*<sup>8</sup> für Python und *Express*<sup>9</sup> für JavaScript.<sup>10</sup>

Die Sprache *JavaScript* wird hier zusammen mit dem *Express*-Framework gewählt, weil die Benutzeroberfläche in JavaScript verfasst werden muss und eine einheitliche Programmiersprache bei paralleler Entwicklung der beiden Komponenten effektiver ist.

Der *Express*-Server wird von der Laufzeitumgebung *Node.js*<sup>11</sup> ausgeführt.

### 3.2.2 Frontend

Das Frontend einer Webanwendung besteht im wesentlichen aus den drei Technologien HTML, CSS und JavaScript. Der HTML-Code gibt die Struktur einer Webanwendung in Form von Objekten wie Absätzen, Links oder Bildern vor und enthält alle statischen Inhalte. Mit CSS lässt sich beschreiben, wie diese auf dem Bildschirm dargestellt werden sollen. Mit JavaScript wird vom Browser ausführbarer Code geschrieben, mit dessen Hilfe eine statische HTML-Seite um dynamische Inhalte und Funktionen ergänzt werden kann.

---

<sup>6</sup><https://python.org>

<sup>7</sup><https://javascript.com>

<sup>8</sup><https://djangoproject.com>

<sup>9</sup><https://expressjs.com>

<sup>10</sup>Ein ausführlicher Vergleich der beiden Frameworks ist unter folgendem Link zu finden <https://dzone.com/articles/nodejs-vs-djangois-javascript-better-than-python>

<sup>11</sup><https://nodejs.org>

#### Framework

Ein Frontend-Framework ist eine JavaScript-Bibliothek, die diese drei Technologien verbindet und die Möglichkeit eröffnet, neben dynamischen Inhalten die gesamte Struktur der Seite durch das Erzeugen, Verändern oder Löschen von HTML-Objekten zur Laufzeit zu verändern. Unter den Frontend-Frameworks gibt es drei große Konkurrenten:

- **Angular**<sup>12</sup>, ein von Google entwickeltes, 2010 veröffentlichtes Framework,
- **React**<sup>13</sup>, ein Framework, das 2013 von Facebook auf den Markt gebracht wurde, und das Framework
- **Vue**<sup>14</sup>, welches im Gegensatz zur Konkurrenz nicht von einer großen Firma, sondern als Open-Source-Projekt mit der Unterstützung einer großen und aktiven Community entwickelt wurde.

Ein ausführlicher Vergleich der drei Frameworks ist in [5] zu finden. Wie dort zu lesen ist, besitzen sowohl *Angular* als auch *React* sehr steile Lernkurven und lassen sich ohne aufwändige Einarbeitung nicht effizient benutzen. *Vue* hingegen überzeugt mit einem einfachen Einstieg und seiner intuitiven Benutzung, welche das Framework für ein Projekt im geplanten Umfang dieser Arbeit ideal macht.

#### Single-Page-Application oder Server-Side-Rendering

Eine Grundüberlegung beim Entwurf einer Webanwendung ist der Aufbau der Seite als Server-Side-Rendering (SSR) oder Single-Page-Application (SPA). Diese beiden Begriffe beschreiben die Art, in der die Webanwendung dem Benutzer bzw. Browser zur Verfügung gestellt wird. Bei einer SSR werden dem Nutzer bei Aufruf eines Links die vollständigen Inhalte der entsprechenden Seite vom Server an den Browser geschickt und dort gerendert. Im Vergleich dazu gibt es bei einer SPA, wie der Name bereits verrät, nur eine einzelne Seite, die dem Benutzer beim ersten Aufruf gesendet wird. Diese enthält das Gerüst der kompletten Anwendung. Wird auf der Seite ein Link geklickt, wird keine neue Seite geladen, sondern lediglich die neuen Teilinhalte vom Server angefordert und angezeigt.

Für Anwendungen wie Online-Shops bieten SSRs den Vorteil einer besseren Suchmaschine-Optimierung und deutlich kürzeren Initial-Ladezeit, welche beide dazu beitragen, Benutzer auf die Seite zu locken. Nach der etwas längeren Lade-

---

<sup>12</sup><https://angular.io>

<sup>13</sup><https://reactjs.org>

<sup>14</sup><https://vuejs.org>

zeit bietet eine SPA dafür ein schnelleres und flüssigeres Benutzererlebnis.

Da das Ziel der hier vorgestellten Webanwendung ein Online-Service ist, der weniger Wert auf das Anlocken von Benutzern als auf einen angenehmen Nutzungs-Flow legt, wird hier eine SPA verwendet.

## 3.3 Kommunikation und Sicherheit

Die Kommunikation zwischen Druckerservern und 3D-Druckern findet über USB statt und ist daher sicher gegenüber digitalen Angriffen. Web- und Druckerserver sind hingegen über das Netzwerk, also ein potenziell unsicheres Medium verbunden. Um trotzdem eine authentifizierte und verschlüsselte Kommunikation zwischen den beiden Instanzen zu ermöglichen, verwenden diese die sichere Version des HTTP-Protokolls, HTTPS.

Die Sicherheit beim HTTPS-Protokoll wird durch einer Zertifikat-Struktur gewährleistet. Hierbei weist sich ein Server gegenüber dem Client mittels eines Zertifikats aus, das von einer Zertifizierungsstelle ausstellt wird. Der Client vertraut wiederum der Zertifizierungsstelle (siehe Abbildung 3.2).

In der im 3D-Druck-Labor verwendeten Konfiguration dient der Webserver als lokale Zertifizierungsstelle (engl. Certificate-Authority (CA)), die für jeden Druckerserver ein HTTPS-Zertifikat signiert. Sendet ein Druckerserver eine mit seinem Zertifikat signierte Nachricht, kann der Webserver diese anhand der Signatur authentifizieren. Die Vertrauenswürdigkeit des Zertifikats ist dabei für den Webserver gegeben, da er es zuvor selbst ausgestellt hat (siehe Abbildung 3.3).

Das Zertifikat dient dabei nicht nur der Authentifizierung, sondern ermöglicht ebenfalls durch die Bereitstellung eines Schlüssels die Verschlüsselung der Kommunikation. Eine ausführliche Beschreibung der Verschlüsselung und digitalen Signatur von Daten findet sich in [3].

Damit auch die Kommunikation des Frontends mit dem Backend des Webserver über HTTPs stattfinden kann, benötigt der Webserver ein Zertifikat einer öffentlichen Zertifizierungsstelle z.B. Let's Encrypt<sup>15</sup>.

---

<sup>15</sup><https://letsencrypt.org>

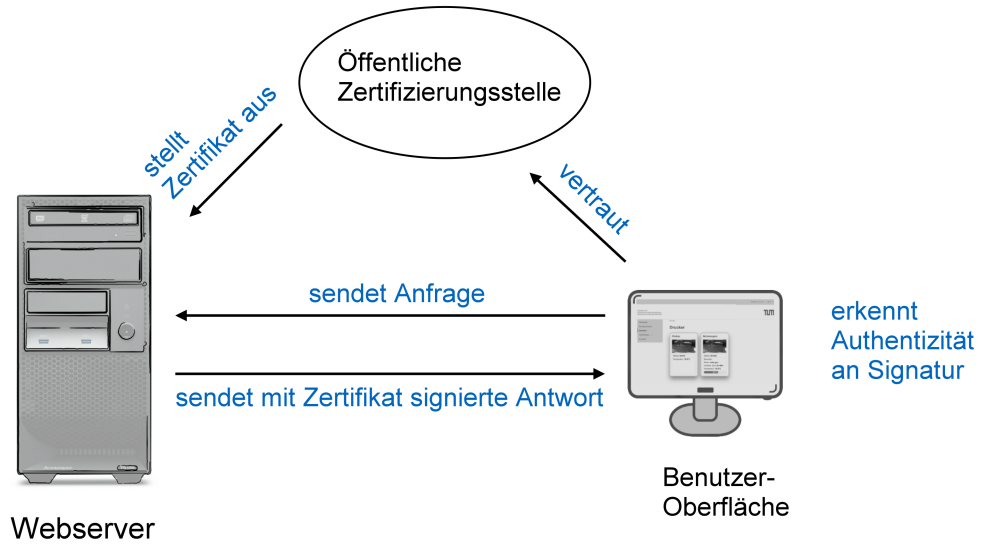


Abbildung 3.2: Die Zertifikatstruktur des HTTPS-Protokolls

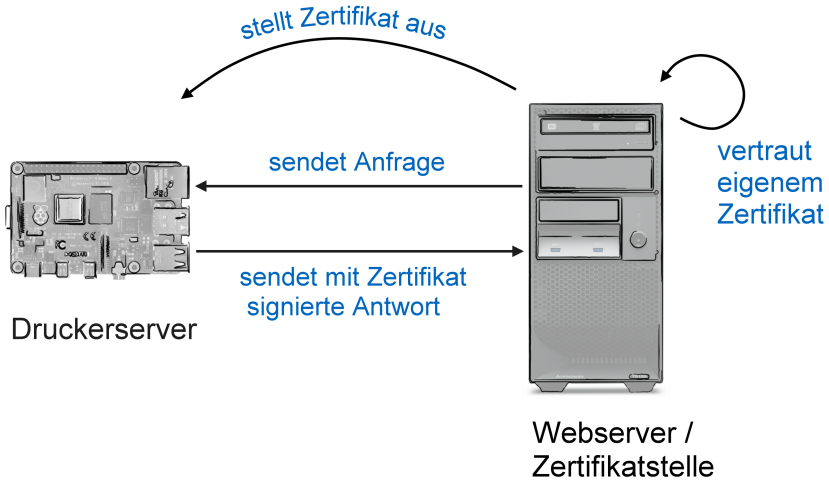


Abbildung 3.3: Die Zertifikatstruktur im 3D-Druck-Labor



## 4 Funktionsweise der Webanwendung

Die Webanwendung kontrolliert den Informationsfluss zwischen Benutzer und 3D-Druck-Labor. Im Folgenden Kapitel wird die Steuerung der Druckerserver durch den Webserver, die dadurch ermöglichte Automatisierung der 3D-Druck-Prozesse und die Funktionalitäten der Benutzeroberfläche beschrieben

### 4.1 Kommunikation zwischen Web- und Druckerserver

Der Druckerserver empfängt Befehle und Status-Anfragen vom Webserver mittels einer Programmierschnittstelle, welche über den HTTP- oder HTTPS-Port zu erreichen ist. Um automatische Push-Benachrichtigungen über den aktuellen Drucker-Status zu erhalten, stellt die *Octoprint* Software einen Websocket bereit. Die Funktionsweise dieser beiden Komponenten wird im Folgenden genauer betrachtet.

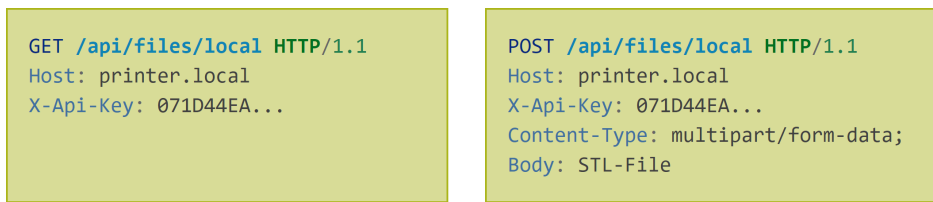
#### 4.1.1 Programmierschnittstelle

Die Programmierschnittstelle (engl. Application Programming Interface (API)) von *Octoprint* verwendet das HTTP-Protokoll, um die Steuerung des 3D-Druckers zu realisieren. Die *Octoprint*-API ist nach dem REST-Standard ("Representational State Transfer") entworfen. Das bedeutet, dass sich die folgende Beschreibung der API auch auf andere REST-APIs, wie z.B. die *Repetier-Server*-API, übertragen lässt.

Eine Anfrage an die API über das HTTP-Protokoll, im Folgenden API-Request genannt, besteht aus einem Pfad, einer Methode und eventuell einem Body, in welchem zu übertragende Daten gespeichert werden. Um die API-Request zu authentifizieren, verwendet *Octoprint* einen API-Schlüssel, welcher in die Request eingefügt werden muss.

In der Dokumentation der *Octoprint*-Software ist ausführlich beschrieben, über welchen Pfad sich welche Funktionen der API ausführen lassen. Die für einen bestimmten Pfad zu verwendende Methode ist fast immer eine der beiden Standard-

## 4 Funktionsweise der Webanwendung



**Abbildung 4.1:** Beispiel für eine GET- und eine POST-Request

HTTP-Methoden GET zum Abrufen von Informationen oder POST zum Senden von Daten. Im Falle einer POST-Request muss auf den Request-Header (Pfad, API-Schlüssel, etc.) ein Body mit den zu verarbeitenden Informationen (z.B einer STL-Datei) folgen. Der Aufbau des Bodies wird ebenfalls in der Dokumentation beschrieben. Eine Darstellung zweier solcher Requests ist in Abbildung 4.1 zu sehen.

Der *Express*-Server verwendet eine Klasse, um Daten vom *Octoprint*-Server abzufragen oder an diesen zu senden. Die Klassen-Methoden implementieren Datei-Operationen wie Upload, Slicen, Drucken und Löschen, das Setzen der Zieltemperaturen für die Temperatursteuerung von Druckkopf und Heizbett und das Senden von Gcode-Befehlen z.B. für das Anheben der Z-Achse.

### 4.1.2 Websocket

Ein Websocket (WS) ist ein Web-Protokoll, welches eine offene, bidirektionale Verbindung zwischen Server und Client darstellt. Der *Octoprint*-Server nimmt in der Socket-Verbindung die Rolle des Servers, der Webserver die des Clients ein. Die offene Verbindung erlaubt es dem Web-Server, Status-Meldungen in Form von Push-Benachrichtigungen zu erhalten, ohne auf Praktiken wie Polling<sup>1</sup> zurückgreifen zu müssen.

Um sich mit dem Websocket des Druckerservers zu verbinden, muss der Webserver das folgende Authentifizierungsverfahren durchlaufen:

Der Webserver erzeugt ein Socket-Client-Objekt und initiiert damit einen Verbindungsversuch zum Druckerserver. Wird die Anfrage vom Server erwidert, ist damit der Websocket-Kanal geöffnet, aber noch nicht authentifiziert. Zur Authentifizierung führt der Webserver über die *Octoprint*-API einen passiven Login durch, welcher als Antwort die aktuelle Session-ID erhält. Diese wird vom Webserver in

<sup>1</sup> Polling bezeichnet das kontinuierliche Abfragen auf Status-Änderungen und ist deshalb vergleichsweise ineffizient.

das benötigte Format { 'auth': '\_api:SessionID' } gebracht und über den Websocket gesendet.

Sind die Angaben korrekt, wird damit die Authentifikation abgeschlossen und Push-Benachrichtigungen vom *Octoprint*-Server an den Webserver ausgegeben (siehe Abb. 4.2). Die für die Benutzer und die Steuerung des 3D-Druck-Prozesses relevanten Benachrichtigungen sind

- `current`, welche regelmäßig über den aktuellen Druckerstatus informieren (Druckfortschritt, Temperatur-Werte, etc.),
- `event`, die z.B durch den Upload einer Datei oder ein beendeten Slicing-Prozess ausgelöst werden und helfen, den Kontroll-Fluss auf dem Webserver zu steuern oder
- vom Typ `slicingProgress` und enthalten den aktuellen Fortschritt eines vom *Octoprint*-Server durchgeführten Slicing-Prozesses.

## 4.2 Automatisierung des 3D-Druck-Prozesses

Um einen Druck zu starten, erwartet das Backend zunächst eine 3D-Datei und einen Satz Slicer-Parameter von der Benutzeroberfläche. Die Datei wird über die API weiter an den *Octoprint*-Server gesendet. Sobald das Hochladen der Datei auf den *Octoprint*-Server abgeschlossen ist, sendet dieser ein `FileUploaded`-Event über den Websocket, woraufhin der Slicing-Prozess gestartet wird. Der gesamte Ablauf eines 3D-Druck-Vorgang ist in Abbildung 4.3 dargestellt und wird im Folgenden erklärt.

### 4.2.1 Slicing-Prozess

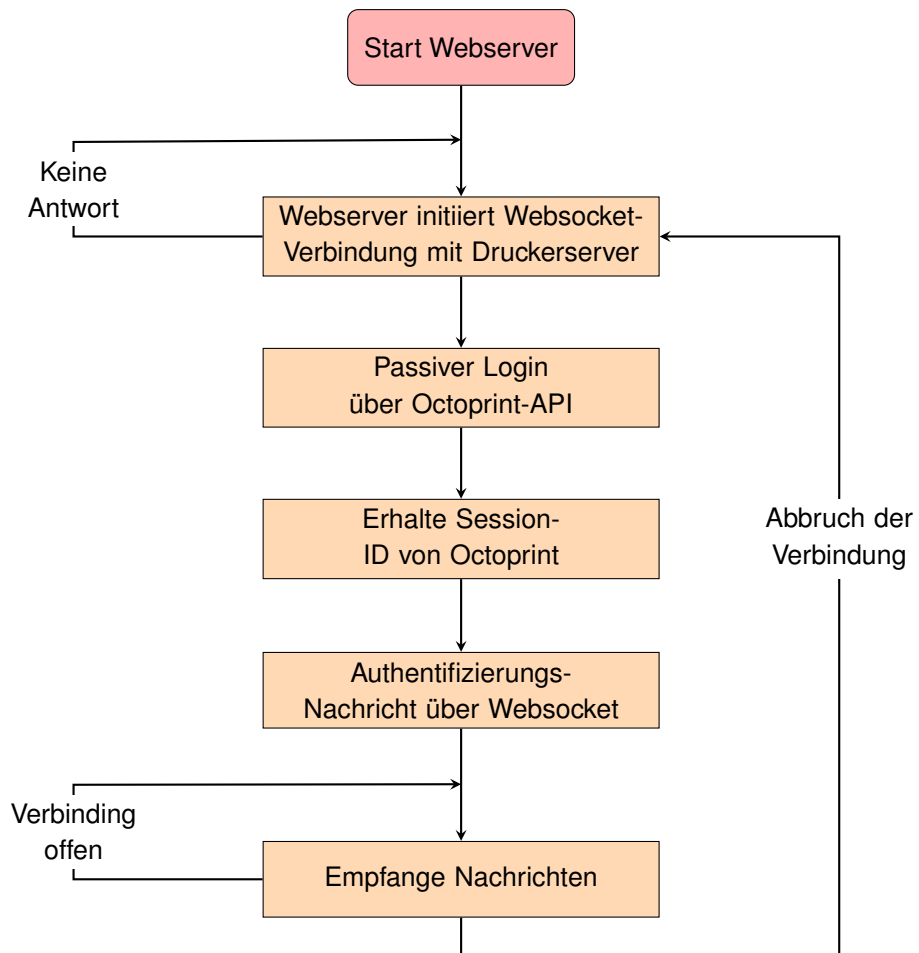
Der Slicing-Prozess findet auf dem Druckerserver statt<sup>2</sup> und wird durch die bereits in der *Octoprint*-API verankerten Funktionen gesteuert. Durchgeführt wird das Slicing der 3D-Datei dabei von der Slicer-Software *Cura*<sup>3</sup>, die auf dem *Octoprint*-Server installiert ist.

---

<sup>2</sup>Obwohl es weitaus weniger Aufwand in der Implementierung wäre, das Slicing dem Benutzer zu überlassen und nur druckfertigen Gcode zu akzeptieren, bringt die Integration des Slicing-Prozesses in die Anwendung zwei große Vorteile:

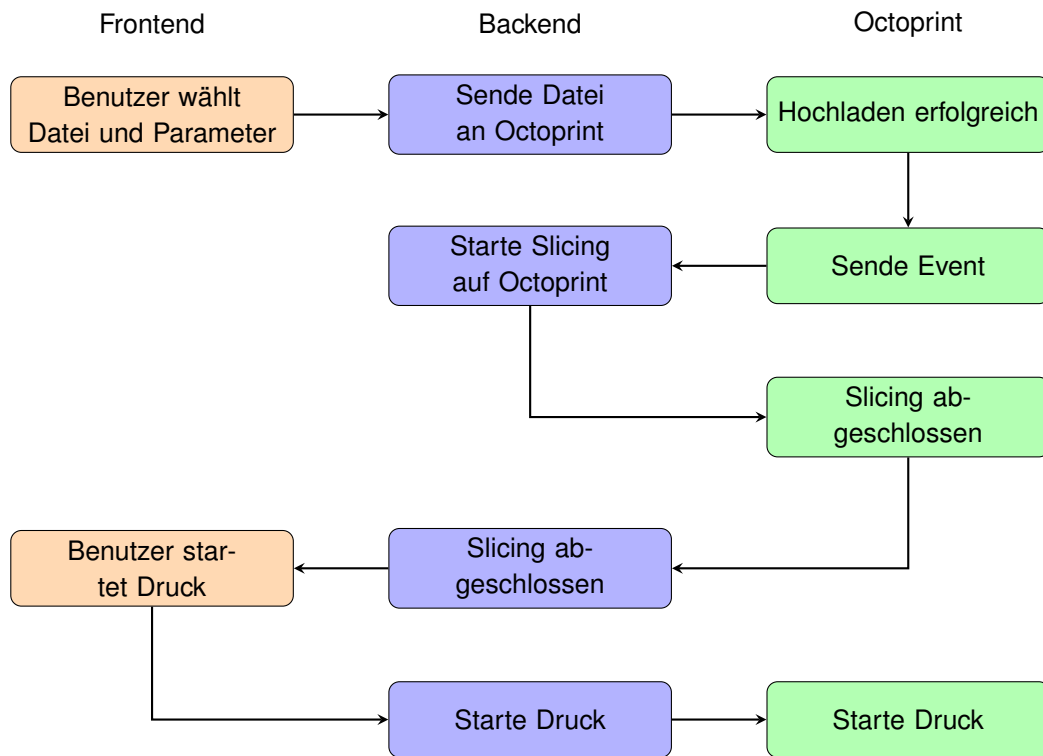
Zum einen verhindert man die Beschädigung der Maschinen durch unabsichtlich oder absichtlich fehlerhaften Gcode. Zum anderen kann man so unerfahrene Benutzer beim komplizierten Slicing-Prozess anleiten.

<sup>3</sup><https://ultimaker.com/software/ultimaker-cura>



**Abbildung 4.2:** Der Authentifikations- und Lebenszyklus des Websockets

## 4.2 Automatisierung des 3D-Druck-Prozesses



**Abbildung 4.3:** Der Ablauf eines 3D-Druck Vorgangs

## 4 Funktionsweise der Webanwendung

Ausgelöst durch das `FileUploaded`-Event sendet der Webserver die API-Request, die den Datei-Befehl zum Slicen enthält. Darin befindet sich die Position und Rotation des Objekts und die zu überschreibenden Slicer-Parameter. Für alle nicht vom Benutzer geänderten Einstellungen verwendet der Slicer ein Standard-Profil, welches für die im Labor verwendeten 3D-Drucker optimiert ist und bereits ohne weitere Modifikationen gute Druckergebnisse erzielt.

Der Fortschritt des Prozesses lässt sich über die Benachrichtigungen vom Typ `slicingProgress` verfolgen. Der erfolgreiche Abschluss wird durch ein entsprechendes Event bestätigt. Der erzeugte Gcode ist auf dem Druckerserver verfügbar und kann dem Benutzer zur Inspektion in einem lokalen Slicer (z.B. *Cura*) bereitgestellt werden.

Auf das Hochladen oder Erzeugen einer Gcode-Datei auf dem Druckerserver folgt eine von *Octoprint* automatisch durchgeführte Analyse des Gcodes. Diese liefert Meta-Informationen über den Druck wie Länge und Volumen des gebrauchten Filaments und die geschätzte Druckzeit.

### 4.2.2 Start des Drucks

Durch einen weiteren Datei-Befehl kann der Druck der erzeugten Gcode-Datei gestartet werden. Damit ist die eigentliche Automatisierung des Labors abgeschlossen, da alle Schritte des 3D-Druck-Prozesses über den Webserver ausgeführt und gesteuert werden können. In der zweiten Hälfte des Kapitels wird die Funktionsweise der Benutzeroberfläche erklärt, die dem Benutzer die oben beschriebenen Funktionen zugänglich macht.

## 4.3 Kommunikation zwischen Webserver und Benutzeroberfläche

Der Webserver muss eine Kommunikations-Schnittstelle für die Benutzeroberfläche bereitstellen, um Daten zu senden und Befehle entgegen zu nehmen. Dafür besitzt er eine mit Hilfe des *Express*-Frameworks realisierte Programmierschnittstelle, Benutzerverwaltung und einen Websocket.

### 4.3.1 Programmierschnittstelle

Wie der *Octoprint*-Server lässt sich der Webserver von einer Programmierschnittstelle (engl. Application Programming Interface (API)) aus erreichen. Durch das *Express*-Framework lassen sich API-Funktionen in den Webserver einbinden, die bei der Umsetzung der Benutzerverwaltung, der Websocket-Authentifikation und der Frontend-Funktionen verwendet werden. Auf der Server-Seite einer API lassen sich die eingehenden Requests außerdem von sogenannter Middleware bearbeiten. Diese speziellen Funktionen modifizieren eingehende HTTP-Requests. So kann eine Middleware z.B. vor der eigentlichen Bearbeitung der Anfrage prüfen, ob eine Request von einem angemeldeten Benutzers kommt und sie andernfalls blockieren.

### 4.3.2 Benutzerverwaltung

Ein essentieller Bestandteil der Frontend-Schnittstelle ist die Benutzerverwaltung, da z.B. alle Routen der Druckersteuerung nur von authentifizierten Benutzern erreicht werden dürfen. Die Funktionsweise der Benutzer-Authentifikation wird im Folgenden erklärt.

Kommt eine Login-Request mit einem Benutzernamen und Passwort über die API, so muss die Identität überprüft werden. Hierfür gibt es eine Benutzerdatenbank, gegen welche die eingehende `login`-Request verglichen wird.

Falls die Überprüfung der Login-Daten erfolgreich verläuft, muss der neue Status des Benutzers gespeichert werden. Hierfür werden Sitzungen (engl. Sessions) verwendet. Eine Session läuft folgendermaßen ab:

Erreicht eine Request den Webserver, so wird diese zuerst an eine Middleware gegeben, welche die Request nach einem Cookie scannt, der eine Session-ID enthält. Wird kein gültiger Cookie gefunden, so generiert die Middleware eine neue ID und damit einen neuen Cookie, der der Antwort an den Browser angehängt wird. Der Browser speichert den Cookie und sendet ihn ab dann bei jeder weiteren Request mit.

Wird hingegen ein Session-ID-Cookie gefunden, so wird die entsprechende Session aus dem Session-Speicher geladen und der Request angehängt. Die Session selbst ist lediglich ein Objekt, in dem Informationen wie der Authentifizierungsstatus des Benutzers gespeichert werden können.

### 4.3.3 Websocket

Wie beim *Octoprint*-Server dient hier der Websocket dazu, Push-Benachrichtigungen - diesmal vom Backend zum Frontend - zu schicken. Der serverseitige Code für die Socket-Verbindung zum Frontend ist etwas komplizierter, da ein Benutzer viele Verbindungen (mehrere Tabs oder Fenster) öffnen und für jede Verbindung Informationen von verschiedenen Druckern anfordern kann. Der Webserver muss also eingehende Öffnungsversuche authentifizieren und dann die Push-Benachrichtigungen von den Drucker-Servern an die richtigen Frontend-Sockets weiterleiten. Die Lösung dieser beiden Probleme wird im Folgenden erläutert.

#### Authentifikation

Die Authentifikation des Websockets gegenüber dem Backend orientiert sich stark an der des *Octoprint*-Websockets. Wird eine neue Verbindung geöffnet, so wartet der Webserver auf eine Authentifikations-Nachricht über den Websocket. Diese Nachricht muss einen in der Nutzerdatenbank enthaltenen Benutzernamen und die dazugehörige Session-ID enthalten. Hierfür ist es erforderlich, dass der Webserver dem Frontend eine Möglichkeit gewährt, die aktuelle Session-ID abzurufen.

Die Bereitstellung der Session-ID wird durch eine entsprechende API-Funktion umgesetzt. Wird vom Frontend eine Session-Request gesendet, so enthält die Antwort des Webserver die aktuelle Session samt Session-ID. Obwohl die in der Session gespeicherten Daten sensibel sind, stellt dieses Verfahren kein Risiko dar, da der Benutzer zuerst seine Authentizität durch den Cookie beweisen muss.

Bei erfolgreicher Authentifikation speichert der Webserver die Socket-Verbindungsdaten zusammen mit dem Benutzernamen und der Information, von welchem Drucker Nachrichten über den Socket empfangen werden sollen, in einer Frontend-Instanz. Diese wird nun an den Socket-Linker übergeben.

#### Socket-Linker

Der Socket-Linker ist die Komponente, die dafür verantwortlich ist, die beim Webserver eingehenden Socket-Nachrichten von allen Druckern an die richtigen Frontend-Instanzen weiterzuleiten. Dafür muss sie speichern, welcher Benutzer in welchen Browser-Tabs oder -Fenstern welche Drucker-Informationen sehen möchte. Wird eine Socket-Verbindung geschlossen, löscht der Socket-Linker zuerst alle Referenzen auf die Frontend-Instanz und danach die Instanz selbst.



### Drucker-Reservierung

Für einen geregelten Betrieb der Anwendung muss sicher gestellt werden, dass nie mehr als ein Nutzer gleichzeitig einen Drucker bedienen kann. Zu diesem Zweck gibt es in der API eine Funktion, mit der ein autorisierter Benutzer einen Drucker aus- oder abwählen kann. Die Auswahl erfolgt nur dann, wenn der Drucker gerade nicht durch einen anderen Benutzer in Gebrauch ist. Wird vom Benutzer ein Druck gestartet, so bleibt der Drucker reserviert, bis der Benutzer bestätigt, das Druckobjekt aus der Maschine entfernt zu haben, und somit den Drucker wieder frei gibt.

Eine grundsätzliche Regel bei der Web-Entwicklung ist es, im Backend keiner Nachricht vom Frontend zu vertrauen, da diese mit potentiell böswilligen Absichten manipuliert werden kann. Konkret bedeutet dies für die Druckerauswahl, dass man sich nicht auf den Nutzer verlassen kann, einen nicht verwendeten Drucker auch wieder abzuwählen. Stattdessen kann z.B. einfach ein Browser-Fenster geschlossen werden. Es wird also ein Verfahren benötigt, um die Aktivität eines Benutzers zu verfolgen und gegebenenfalls die betroffenen Drucker wieder freizugeben.

Eine Lösungsmöglichkeit wäre, bei jedem Zugriff auf einen Drucker die momentane Zeit zu speichern und den Drucker nach einer festen Zeit wieder freizugeben. Dieses Verfahren ist allerdings nicht besonders benutzerfreundlich. Eine elegantere Lösung des Problems ist die Verwendung der bereits schon funktionalen Websockets. Der Web-socket registriert verlässlich das Schließen des Kanals, sobald die Gegenseite die Kommunikation abbricht. Um die Websockets zur Aktivitätserkennung zu nutzen, muss der oben beschriebene Socket-Linker zusätzlich zu jedem Benutzer speichern, wie viele Verbindungen dieser geöffnet hat. Sobald die letzte Verbindung geschlossen wird, kann er alle nicht druckenden Drucker freigeben.

## 4.4 Frontend

Das Frontend, also die Benutzeroberfläche der Webanwendung, bietet dem Nutzer die zuvor im Backend implementierten Funktionen zur Druckersteuerung und enthält Informationen und Hilfestellung zum 3D-Druck-Prozess und zur Benutzung der Webanwendung.

### 4.4.1 Basis-Layout

Das Layout wurde nach den Vorgaben des *TUM Corporate Design* umgesetzt. Im Style-Guide wird die HTML-Struktur beschrieben, welche zusammen mit dem zur

## 4 Funktionsweise der Webanwendung



**Abbildung 4.4:** Das TUM-Layout mit Header, Navigation, Inhalt und Footer



**Abbildung 4.5:** Die Funktionsleiste des Headers im ab- und angemeldeten Zustand

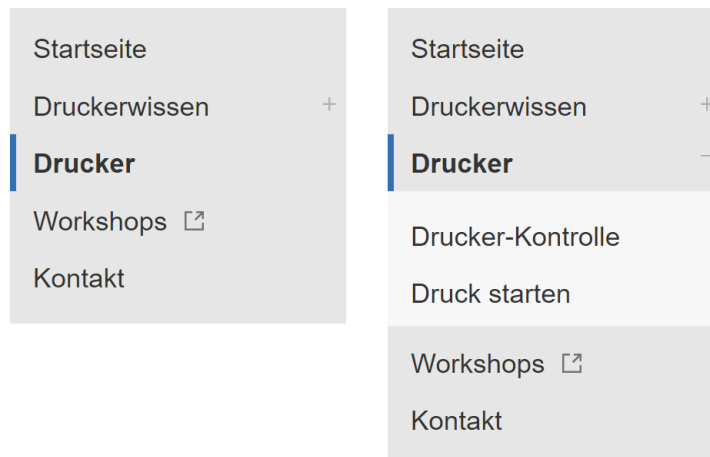
Verfügung gestellten CSS-Stylesheet das Aussehen der Seite bestimmt. Das Erscheinungsbild im Corporate Design hilft der Anwendung, sich in andere Online-Auftritte der TUM einzugliedern. Das Layout besteht aus einem Header, einer Navigationsspalte, einem Inhaltsbereich und einem Footer (siehe Abbildung 4.4).

### 4.4.2 Komponenten

#### Header

Der Header dient in erster Linie dazu, der Seite das charakteristische Aussehen der TUM-Webseiten zu verleihen. Die Funktionsleiste am oberen Rand enthält außerdem Informationen über den Anmeldestatus des Benutzers und dementsprechend einen Link zur Anmelde-Seite oder einen Knopf zum Abmelden.

Der Header hat des Weiteren die Aufgabe, die Benutzeraktivität zu verfolgen und den Anmeldestatus in allen Tabs zu synchronisieren. Die Aktivität eines Benutzers wird im Backend durch die Anzahl der offenen Socket-Verbindungen bestimmt. Der



**Abbildung 4.6:** Die Navigationsspalte ohne und mit ausgewähltem Drucker

Header öffnet deswegen eine Websocket-Verbindung, so dass für jedes offene Tab mindestens eine Verbindung aktiv ist, unabhängig davon, ob der Benutzer gerade Informationen über einen Drucker erhält oder nicht. Der Websocket wird geschlossen, sobald der Benutzer sich abmeldet oder das Tab schließt. Über den lokalen Speicher des Browsers kann der Anmeldestatus zwischen Tabs synchronisiert werden. Dazu muss das Tab, in dem der Anmeldevorgang stattgefunden hat, die Daten des Nutzers in den Speicher schreiben. Ein in jedem Tab aktives Beobachter-Objekt registriert die Veränderung und aktualisiert den Anmeldestatus.

### Navigationssleiste

Im Style Guide wird die HTML-Struktur der Navigation für jede Seite angepasst, so dass aktive Seiten markiert und eventuelle Untermenüs angezeigt werden. Für eine herkömmliche Seite, die *Server Side Rendering* verwendet, ist dies durchaus eine Standard-Vorgehensweise. Dagegen gibt es bei einer, hier verwendeten, *Single-Page-Application* nur eine Seite, weswegen die HTML-Struktur nicht durch den Entwickler, sondern durch zusätzliche Logik verändert werden muss. Mit Hilfe der Funktionen des *Vue*-Frameworks lassen sich Untermenüs dynamisch ein- und ausblenden. Zusätzlich werden auch die Unterseiten der Druckersteuerung verborgen, bis ein Drucker ausgewählt ist.

## 4 Funktionsweise der Webanwendung

The screenshot shows the login page of a printer control web application. At the top right, there is a button labeled 'Anmelden'. Below this, on the left, is the header for 'Daedalus 3D', 'Lehrstuhl für Datenverarbeitung', and 'Technische Universität München', along with the TUM logo. A navigation menu on the left lists 'Startseite', 'Druckerwissen +', 'Drucker' (highlighted), 'Workshops ↗', and 'Kontakt'. The main content area is titled 'Drucker' and 'Anmeldung'. It contains a login form with two input fields: 'Benutzer' (containing 'matthias') and 'Passwort' (containing '....'). Below the fields is a blue 'Anmelden' button.

**Abbildung 4.7:** Die Startseite der Druckersteuerung mit ausgefülltem Anmeldeformular

### Inhaltsbereich

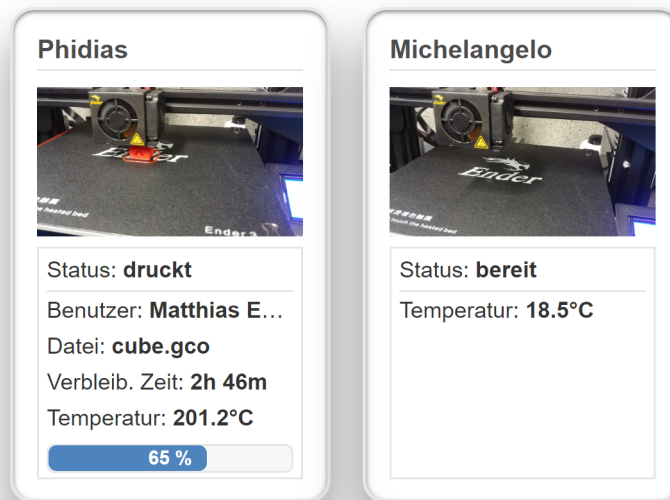
Der Inhaltsbereich enthält alle Seiten der Webanwendung. Diese bestehen aus der Druckersteuerung und den statische Seiten, die nützliches Wissen über 3D-Drucker und deren Bedienung enthalten. Im Folgenden wird die Verwendung und Funktionsweise der Oberfläche zur Druckersteuerung erläutert.

Da der *TUM Style Guide* keine Anwendungen zur 3D-Druck-Steuerung vorsieht, ist das Design der entsprechenden Seiten selbst definiert und auf Übersichtlichkeit und Benutzerfreundlichkeit ausgelegt.

**Die Druckersteuerung** ist unter dem Menü-Punkt *Drucker* zu erreichen und durch die Benutzerzugriffskontrolle geschützt. Wird wie in Abbildung 4.7 ein Login-Versuch über das Login-Formular durchgeführt, werden Benutzername und Passwort an die API des Backends geschickt und dort geprüft.

Ist der Login-Versuch erfolgreich, so wird die Druckerübersicht gezeigt (siehe Abbildung 4.8). Hier wird für jeden Drucker eine Karte mit amüsanten, in Echtzeit aktualisierten Informationen angezeigt. In der Übersicht kann ein Administrator die

## Drucker



**Abbildung 4.8:** Die Druckerübersicht mit Live-Informationen

Aktivität im ganzen Labor überwachen und sehen, wer gerade welchen Drucker benutzt. Jede Drucker-Karte ist außerdem ein Link, der die Steuerung des ausgewählten Druckers öffnet bzw. diesen für den angemeldeten Benutzer reserviert, sofern der Drucker aktuell verfügbar ist.

Die Liste der Drucker erhält das Frontend durch eine Request an die Backend-API. Für die in der Liste enthaltenen Drucker wird dann eine Websocket-Verbindung geöffnet. Informationen wie Druckkopf-Temperatur und Druckfortschritt werden dadurch in der Druckerübersicht in Echtzeit aktualisiert.

Hat der Benutzer durch einen Klick auf eine der Karten einen Drucker ausgewählt, wird dieser mittels einer API-Request reserviert und die beiden Seiten *Drucker-Kontrolle* und *Druck starten* im Menü zur Verfügung gestellt.

**Die Drucker-Kontrolle** dient dazu, ausgewählte Drucker-Module fernzusteuern (siehe Abbildung 4.9). Die Temperatur-Kontrolle ist nützlich, um z.B. für einen Filamentwechsel den Druckkopf vorzuheizen oder während eines laufenden Druckvorgangs Korrekturen an der Drucktemperatur vorzunehmen.

#### 4 Funktionsweise der Webanwendung

##### TEMPERATUR

	Ist	Soll		
Extruder-Temperatur	18.3°C	<input type="text" value="0"/> °C	<input type="button" value="PLA vorheizen"/>	<input type="button" value="Abkühlen"/>
Heizbett-Temperatur	18.1°C	<input type="text" value="0"/> °C	<input type="button" value="PLA vorheizen"/>	<input type="button" value="Abkühlen"/>

##### BEWEGUNG

Z-Achse anheben (mm):

**Abbildung 4.9:** Die Drucker Kontroll-Funktionen

Die Funktion zum Anheben der Z-Achse ist wichtig, um z.B. im Falle eines gescheiterten Drucks die heiße Düse vom Druck-Objekt zu entfernen. So können Schäden am Druckkopf verhindert werden.

**Die Druck starten-Seite** ist das Kernstück der gesamten Anwendung. Der Benutzer hat hier die Möglichkeit, Änderungen an den Slicer-Parametern vorzunehmen und eine STL-Datei für den Druck hochzuladen. Welche Parameter welchem Nutzer zur Verfügung stehen, kann der Administrator der Anwendung bestimmen. Eine beispielhafte Einteilung der Parameter in Profile für Benutzer unterschiedlicher Fertigungslevel folgt in Kapitel 5.1.

Sind die gewünschten Slicer-Einstellungen eingegeben, muss ein 3D-Objekt zum Hochladen ausgewählt werden. Die Datei wird dann zusammen mit den Slicer-Einstellungen an das Backend geschickt. Von dort wird die Datei an den *Octoprint*-Server weitergeleitet. Ist der Upload abgeschlossen, sendet der Webserver automatisch den Befehl zum Slicen. Für den Benutzer öffnet sich nach Auswählen der Datei ein Pop-Up-Fenster, in dem der Fortschritt des Slicing-Prozesses angezeigt wird. Ist das Slicing beendet, so werden Meta-Informationen über den Druck wie geschätzte Druckdauer und verbrauchtes Filament angezeigt. Der Benutzer hat jetzt die Möglichkeit, direkt den Befehl *Druck starten* zu geben, die vom Slicer erzeugte Gcode-Datei zur Inspektion herunterzuladen oder den Druckversuch abzubrechen.

SLICER SETTINGS		SLICER SETTINGS	
Layer Höhe:	<input type="text" value="0,2"/> mm	Layer Höhe:	<input type="text" value="0,15"/> mm
Füllgrad:	<input type="text" value="20"/> %	Wanddicke:	<input type="text" value="1,2"/> mm
Druckbett Temperatur:	<input type="text" value="60"/> °C	Druckgeschwindigkeit:	<input type="text" value="65"/> mm/s
Druckkopf Temperatur:	<input type="text" value="200"/> °C	Füllgrad:	<input type="text" value="15"/> %
Support:	<input type="text" value="Kein Support"/>	Druckbett Temperatur:	<input type="text" value="56"/> °C
		Druckkopf Temperatur:	<input type="text" value="204"/> °C
		Support:	<input type="text" value="Nur von Druckplattform"/>
		Druckbetthaftung:	<input type="text" value="Raft"/>

Abbildung 4.10: Die Slicer-Einstellungen mit Profilen für Anfänger und Fortgeschrittene

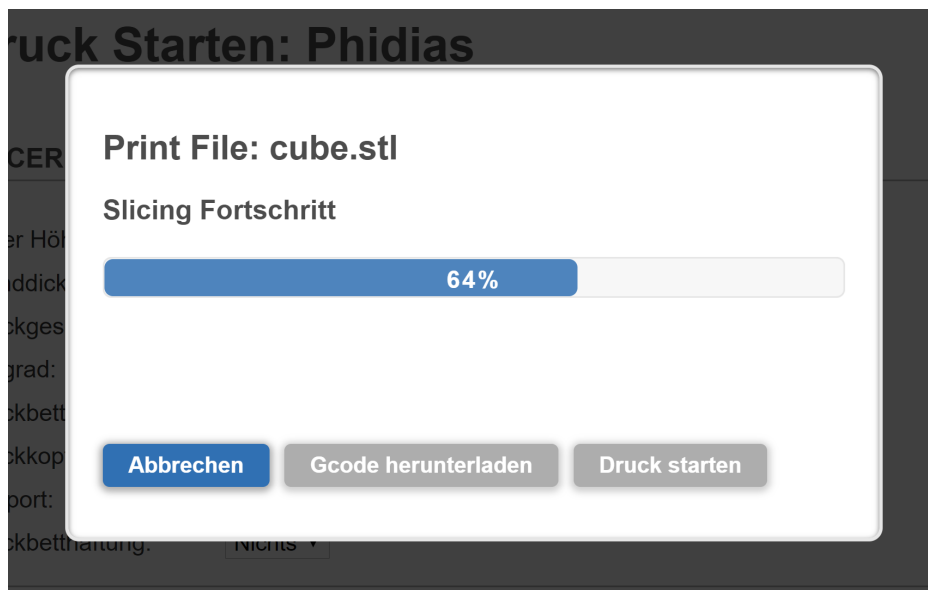
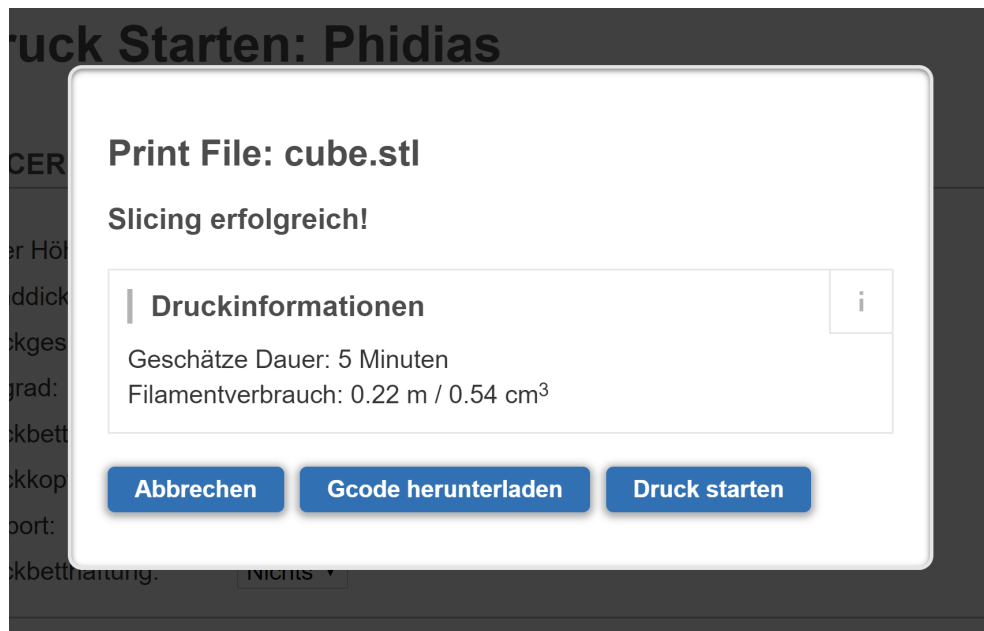


Abbildung 4.11: Das Pop-Up mit der Anzeige für den Slicing-Fortschritt



**Abbildung 4.12:** Die Meta-Informationen zum druckfertigen Objekt

### Footer

Der Footer entspricht dem *TUM Style Guide* und erfüllt für die Webanwendung keine besondere Funktion.



## 5 Anwendung im 3D-Druck-Labor

Das 3D-Drucker-Labor wurde vom *Lehrstuhl für Datenverarbeitung (LDV)* aufgebaut, um Studenten mit der Technologie des 3D-Druckers vertraut zu machen. Der Einstieg in den Umgang mit 3D-Druckern erfolgt mittels vom *LDV* veranstalteter Workshops. Um eigene Erfahrungen zu sammeln und Projekte umzusetzen, haben die Studenten Gelegenheit, im Labor mit 3D-Druckern zu arbeiten.

Die in dieser Arbeit entwickelte Anwendung zur Automatisierung des Labors gewährt dem Administrator des Labors vollständige Kontrolle über den 3D-Druck-Prozess und insbesondere das Slicen der 3D-Objekte innerhalb der Anwendung. Dies erlaubt es, die Studenten im Lernprozess zu begleiten und sie erst nach und nach mit den vielen Einstellmöglichkeiten zu konfrontieren. In diesem Kapitel werden einige Anwendungsgebiete vorgestellt, für welche die Software verwendet werden kann.

### 5.1 Benutzerprofile

Ein Administrator kann beliebig viele Profile für die Benutzer des 3D-Drucker-Labors definieren, z.B. „Anfänger“, „Fortgeschritten“, „Profi“. Ein Profil enthält alle Berechtigungen und Einschränkungen, die einem Benutzer aufgrund seines Fertigungsgrads zugeordnet werden sollen. Mögliche Inhalte eines solchen Profils sind:

- Typ und Anzahl der Drucker, die benutzt werden dürfen
- verschiedene Arten von Filament, welche für den Druck verwendet werden können (PLA, ABS, FLEX, etc.)
- Slicer-Einstellungen, deren Werte der Benutzer verändern darf (siehe Tabelle 5.1)
- Intervalle, in denen bestimmte Parameter gesetzt werden dürfen (z.B. Temperatur des Druckkopfs nur im Bereich von 200 °C bis 210 °C)
- maximale Druckzeit pro Druck oder eine maximale Druckzeit pro Tag

Des Weiteren lassen sich die Profile auch für konkrete Projekte wie die vom *LDV* ver-

## 5 Anwendung im 3D-Druck-Labor

anstalteten Workshops nutzen. Dabei könnten die Profile z.B. die für eine Aufgabe benötigten Einstellungen beinhalten. Nach Abschluss eines Workshops kann den Studenten dann ein neues Profil zugewiesen werden, mit dem sie alle neu erlernten Features anwenden können.

Slicer-Parameter	Auswirkungen	Risiken
Schichthöhe	Detailgrad Druckdauer	zu klein → Unter-Extrusion zu groß → Extruder überlastet
Druckkopf-Temperatur	Schichtbindung Überhänge	Zu kalt → Verstopfung Zu heiß → Konturen verlaufen
Filamenteinzug	Stringing	zu lang → Flüssiger Kunststoff in kaltem Bereich kann Hotend verstopfen
Füllgrad	Stabilität Qualität der Oberseite	Verschwendung von Filament

**Tabelle 5.1:** Auswirkungen und Risiken einzelner Slicer-Parameter

## 5.2 Protokollierung und Auswertung

Zur Kontrolle des Labors lassen sich alle Druckvorgänge, die über den Webserver durchgeführt werden, protokollieren und statistisch auswerten. So können Verhaltensmuster erkannt und für die Verbesserung des didaktischen Konzepts bei der Vermittlung von 3D-Druck-Inhalten eingesetzt werden.

Gleichzeitig ist ein System denkbar, in dem ein Student bestimmte Errungenschaften erreichen kann, um dadurch sein Können zu beweisen und z.B. neue Slicer-Parameter freizuschalten (Gamification).

## 5.3 Kontrolle der Drucker

Durch die globale Erreichbarkeit des Webserver wird die Kontrolle der laufenden 3D-Drucke-Prozesse durch den Benutzer ermöglicht. 3D-Drucke können, sofern von den Administratoren so vorgesehen, von überall gestartet, überwacht und gegebenenfalls angepasst oder abgebrochen werden. Neben der manuellen Überwachung bietet die auf dem Webserver geschaffene Plattform auch die Möglichkeit, eine automatische Fehlererkennung wie in den Artikeln [1] und [6] beschrieben zu implementieren.

## 6 Fazit und Erweiterungen für den Betrieb

In der Motivation der Arbeit wurde erörtert, warum es für Studenten der Ingenieurwissenschaften wichtig ist, Erfahrungen mit 3D-Druckern zu sammeln und warum die Bereitstellung von 3D-Druckern ohne eine geeignete Infrastruktur nicht praktikabel ist. Als Ziel wurde deswegen die Aufgabe gestellt, Möglichkeiten zur Automatisierung des 3D-Druck-Prozesses zu untersuchen und einen als geeignet empfundenen Ansatz testweise umzusetzen.

Die für die Umsetzung der Automatisierung gewählten Hard- und Softwarekomponenten erfüllen alle Anforderungen und ermöglichen die zentrale Steuerung und Verwaltung der 3D-Drucker durch die im Rahmen der Arbeit entwickelte Webanwendung.

Über die Benutzeroberfläche der Webanwendung wird die Bedienung eines 3D-Druckers für den Benutzer stark erleichtert. Gerade unerfahrene Nutzer können so einen schnellen Einstieg in das Feld des 3D-Druck finden und durch die in Kapitel 5.1 vorgestellten Benutzerprofile graduell mit komplexeren Aspekten der Technologie konfrontiert werden. Durch die Einhaltung des *TUM Style Guides* finden sich Studenten der *TUM* schnell zurecht und die Anwendung lässt sich natlos in die Web-Präsenz der *TUM* einfügen.

Um die Stabilität und Skalierbarkeit der Anwendung für den Einsatz im 3D-Druck-Labor zu verbessern, müssten einige Funktionalitäten, die bisher aus Zeitgründen in vereinfachter Form umgesetzt sind, erweitert werden.

### 6.1 Datenbank

Die Datenbank enthält Tabellen mit Informationen über

- die Drucker (IP-Adresse, API-Schlüssel, etc.)
- die Benutzerprofile (Parameter, zulässige Intervalle)

## 6 Fazit und Erweiterungen für den Betrieb

- die Benutzer (Slicer-Profil, aktuell genutzte Drucker, etc.)
- die Slicer-Parameter
- protokollierte Vorgänge
- die verfügbaren Filamente (Farbe, Preis, etc.)

Der Einfachheit halber besteht die Datenbank bisher aus einer Datei. Stattdessen kann an dieser Schnittstelle jedoch eine beliebige Datenbank wie z.B. die eines Datenbank-Service-Anbieters wie *MongoDB*<sup>1</sup> angeschlossen werden. Damit wäre die Anwendung skalierbar und könnte von einer beliebig großen Anzahl an Benutzern genutzt werden.

### 6.2 Authentifikation

Die Authentifikation ist noch in einfacher Form, aber mit einer klaren Schnittstelle umgesetzt. Um die Identität eines Benutzers zu verifizieren, werden Benutzername und Passwort mit dem entsprechenden Eintrag in der Datenbank verglichen. Die Identitätsprüfung könnte schnell durch ein anders Verfahren ersetzt werden.

Die optimale Lösung wäre hier eindeutig, die Datenbank der TUM zu nutzen, welche über einen Service namens *shibolet*<sup>2</sup> die Rolle eines Identity-Providers einnehmen kann. So würde es den Studenten ermöglicht, sich mittels ihrer LRZ-Kennung zu authentifizieren. Dafür müsste sich ein Service-Provider, wie die in dieser Arbeit entwickelte Anwendung, auch gegenüber der TUM authentifizieren. Dieser aufwändige Prozess war im Rahmen dieser Arbeit nicht sinnvoll.

### 6.3 Anpassung des Layouts an mobile Endgeräte

Das Layout der TUM-Website, welches der Benutzeroberfläche der Webanwendung zugrunde liegt, ist responsiv, d.h. die Darstellung passt sich bei Verwendung auf einem Tablet oder Smartphone der Größe des Ausgabegeräts an. Die Benutzeroberfläche der Webanwendung ist bisher nur für die Verwendung auf einem Desktop-Monitor vorgesehen. Damit sie auch komfortabel von einem Tablet oder Smartphone aus verwendet werden kann, müssen die CSS-Dateien noch entsprechend ergänzt werden.

---

<sup>1</sup><https://mongodb.com>

<sup>2</sup><https://lrz.de/services/netzdienste/shibolet-aai>

## 6.4 Druckkostenberechnung

Um die im 3D-Druck-Labor entstehenden Kosten mit zu decken und zu verhindern, dass Druckzeit und Filament durch unüberlegte Drucke verschwendet werden, sollen den Studenten die Kosten ihrer 3D-Drucke anteilig in Rechnung gestellt werden. Hierfür würde eine Druckkostenberechnung benötigt.

Die Druckzeit und das benötigte Filament werden beim Slicen eines 3D-Objekt bereits berechnet und ergeben zusammen mit dem Preis des verwendeten Filaments die Kosten für den Druck. Für die Abrechnung müssten die Kosten nur im Konto des Benutzers vermerkt und von ihm eingefordert werden.



# Literaturverzeichnis

1. F. Baumann und D. Roller. Vision based error detection for 3D printing processes. In *International Conference on Frontiers of Sensors Technologies*, 59, Mai 2016. doi:10.1051/mateconf/20165906003.
2. F. Horsch. *3D-Druck für alle*. Carl Hanser Verlag, München, 2014. ISBN 978-3-446-43698-5.
3. C. Paar und J. Pelzl. *Understanding Cryptography*. Springer Verlag, Heidelberg, Dordrecht, London, New York, 2010. ISBN 978-3-642-04100-6.
4. B. Redwood, F. Schöffel und B. Garret. *The 3D Printing Handbook*. 3D Hubs, Amsterdam, Februar 2018. ISBN 978-90-827485-0-5.
5. S. Springer. Angular, vue.js und react: Drei javascript-frameworks im vergleich. In *iX 7/2018*, 2018.
6. J. Straub. Initial work on the characterization of additive manufacturing (3D printing) using software image analysis. In *Machines — Open Access Journal*, 2015. doi:10.3390/machines3020055.