



**IT-BERUFE**

# Protokoll über die betriebliche Projektarbeit

*Diese Erklärung ist der Dokumentation nach dem Deckblatt beizufügen.*

Hiermit versichere ich,

**Till Tomczak**

Vor- und Nachname des Prüfungsteilnehmers

die betriebliche Projektarbeit

**MYP – Manage Your Printer**

Genauere Bezeichnung der Projektarbeit/ Prüfungsausschuss

sowie die eingereichte Dokumentation unter Betreuung von

**Martin Noack / Mercedes-Benz AG**

Vor- und Nachname des Ausbilders/ Name des Ausbildungsbetriebes

**selbstständig** und **ohne fremde Hilfe** konzipiert, verfasst und durchgeführt zu haben. Teile der Dokumentation, die ich **nicht selbstständig** erstellt habe, sind von mir entsprechend **gekennzeichnet** worden. Darüber hinaus erkläre ich, dass ich bei der vorliegenden Dokumentation **keine IT-/KI-gestützten Schreibwerkzeuge** genutzt habe. Die von der Verordnung vorgesehene Richtzeit wurde eingehalten. Die Arbeit hat in dieser Form oder ähnlicher Form keiner anderen Prüfungsinstitution vorgelegen.

Ich bin darüber aufgeklärt worden, dass meine betriebliche Projektarbeit bei **Täuschungshandlungen, bzw. Ordnungsverstößen mit „Null“ Punkten bewertet** wird und als nicht bestanden gilt.

Ich bin weiter darüber aufgeklärt worden, dass dies auch dann gilt, wenn festgestellt wird, dass meine Projektdokumentation im Ganzen oder zu Teilen mit der eines anderen Prüfungsteilnehmers übereinstimmt. Mit der Durchführung einer Plagiatsprüfung und einer Prüfung, ob IT-/KI-gestützte Schreibwerkzeuge eingesetzt wurden, erkläre ich mich einverstanden.

### Arbeitszeit

Das Projekt wurde von mir in der kalkulierten Zeit **komplett fertiggestellt, einschließlich erforderlicher Nacharbeit**  ja  nein

Nein, die Zeit wurde um 2 Stunden  unterschritten  überschritten.

### Fehleranalysen und Einarbeitung in Programmiersprache bestehender Systeme

Begründung

### Persönliche Erklärung

Ich versichere durch meine Unterschrift, dass ich die Projektarbeit und die eingereichte Dokumentation selbstständig angefertigt, alle Stellen, die ich wörtlich oder annähernd wörtlich aus Veröffentlichungen entnommen, als solche kenntlich gemacht habe.

**04.06.2025**

Datum



Unterschrift Prüfungsteilnehmer/-in



Unterschrift Projektverantwortlicher/-r des Auftraggebers

The logo of the IHK (Industrie- und Handelskammer) is displayed. It consists of a dark blue square with the letters 'IHK' in white, bold, sans-serif font. To the left of the square is a light blue curved shape that partially overlaps the square's left edge.

IHK

Abschlussprüfung - Sommer 2025

Fachinformatik für digitale Vernetzung -  
Dokumentation der betrieblichen Projektarbeit

Abgabedatum: 5. Juni 2025

# MYP – Manage Your Printer

Vernetzte 3D-Druck-Reservierungsplattform mit IoT-Anbindung und zentraler  
Verwaltungsoberfläche

## Prüfungsbewerber

Till Tomczak  
Hainbuchenstraße 19  
D-16761 Hennigsdorf

## Ausbildungsbetrieb

Mercedes-Benz AG  
Daimlerstraße 143  
D-12277 Berlin



Mercedes-Benz



## Inhaltsverzeichnis

1. Einleitung.....	1
1.1 Ausgangssituation und Problemstellung.....	1
1.2 Projektziele .....	1
1.3 Projektabgrenzung .....	1
1.4 Projektumfeld und betriebliche Schnittstellen .....	2
2. Projektplanung .....	2
2.1 Zeitplanung nach V-Modell .....	2
2.2 Ressourcenplanung .....	3
2.3 Qualitätssicherungsplanung .....	3
3. Analyse und Bewertung.....	4
3.1 Bewertung der vorhandenen Systemarchitektur.....	4
3.2 Bewertung der heterogenen IT-Landschaft .....	4
3.3 Analyse der IT-sicherheitsrelevanten Bedingungen.....	4
3.4 Anforderungsgerechte Auswahl der Übertragungssysteme .....	5
4. Systemarchitektur und Schnittstellenkonzeption .....	5
4.1 Gesamtsystemarchitektur .....	5
4.2 Technische Systemarchitektur .....	6
4.3 Schnittstellenkonzeption.....	6
4.4 Datenmodell .....	6
5. Umsetzung.....	6
5.1 Implementierung der Backend-Infrastruktur .....	6
5.2 IoT-Integration und Hardware-Steuerung.....	7
5.3 Sicherheitsimplementierung .....	7
5.4 Systemkonfiguration und Deployment .....	8
6. Test und Optimierung.....	8
6.1 Testdurchführung und -ergebnisse .....	8
6.2 Sicherheitstests .....	8
6.3 Systemstabilität und Monitoring.....	9
7. Projektabschluss.....	9
7.1 Soll-Ist-Vergleich.....	9
7.2 Projektergebnisse und Erkenntnisse .....	9
7.3 Optimierungsmöglichkeiten .....	10
7.4 Formale Projektabnahme .....	10
Glossar .....	1
Anlagen.....	1



Übergabeprotokoll .....	1
Screenshots des Projektarbeitsprozesses .....	3
API-Dokumentation .....	4
Benutzerhandbuch .....	4
(geplante) Schulungspräsentation .....	5
Testprotokolle .....	6
Screenshots der Benutzeroberfläche .....	7



## 1. Einleitung

### 1.1 Ausgangssituation und Problemstellung

Die Technische Berufsausbildungsstätte (TBA) der Mercedes-Benz AG am Standort Berlin verfügt über sechs 3D-Drucker verschiedener Hersteller (Prusa, Anycubic), die als wichtige Ressource für die praktische Berufsausbildung dienen. Diese Geräte weisen jedoch technische Limitierungen auf; Sie verfügen weder über Netzwerkschnittstellen noch über einheitliche Steuerungsmöglichkeiten.

Das - wollte man es als ein solches bezeichnen - bestehende Reservierungssystem basierte auf einem analogen Whiteboard, was zu systematischen Problemen führte. Doppelbuchungen traten auf, wenn mehrere Nutzer zeitgleich etwas drucken wollten. Die manuelle Aktivierung und Deaktivierung der Geräte wurde häufig versäumt, was zu unnötigem Energieverbrauch und erhöhtem Verschleiß führte. Eine verlässliche Dokumentation der tatsächlichen Nutzungszeiten existierte nicht, wodurch weder aussagekräftige Betätigungs- und Verantwortungszuordnung, noch eine verursachungsgerechte Kostenzuordnung möglich waren.

Ein vorhandener Frontend-Prototyp des ehemaligen Auszubildenden aus der Fachrichtung der Daten- und Prozessanalyse Torben Haack bot eine moderne Benutzeroberfläche, verfügte jedoch nicht über eine funktionsfähige Backend-Anbindung zur praktischen Nutzung. Diese Ausgangslage bot mir eine fantastische Gelegenheit, im Rahmen meiner Projektarbeit eine weiterführende und meiner Fachrichtung entsprechende Lösung zu entwickeln. Das Projekt weckte – im Gegensatz zu anderen verfügbaren Projektoptionen, die eher einen pflichtgemäßen Charakter hätten, meine intrinsische Motivation deutlich.

### 1.2 Projektziele

Das Projekt "MYP – Manage Your Printer" zielt auf die vollständige Digitalisierung des 3D-Drucker-Reservierungsprozesses durch die Etablierung cyberphysischer Kommunikation mit den relevanten Hardwarekomponenten ab.

Die primären Ziele umfassen die Entwicklung einer webbasierten Reservierungsplattform, die Integration automatischer Hardware-Steuerung via IoT-Komponenten, die Implementierung einer vernetzten, zentralen Verwaltungsoberfläche sowie die Etablierung robuster Authentifizierung und Rechteverwaltung; ganz auch im Sinne der Sicherheit zur Zufriedenheit der IT-Abteilung. Als sekundäre Ziele wurden die Optimierung der Energieeffizienz durch automatisierte Steuerung, die Bereitstellung von Nutzungsstatistiken und Monitoring, die Gewährleistung einer herstellerunabhängigen Lösung sowie die Einhaltung zusätzlicher Sicherheitsrichtlinien definiert.

### 1.3 Projektabgrenzung

Der Projektumfang wurde pragmatisch Umsetzung einer funktionsfähigen Lösung fokussiert; auch wenn der eigene Ehrgeiz weit darüber hinaus ging, musste das Ganze im (zeitlichen) Rahmen bleiben. Primärer Fokus des Projektes war die Entwicklung des Backends - sprich: der funktionalen Vernetzung; die Datenbankanbindung für die Reservierungsverwaltung, die IoT-Integration via Smart-



Plugs, Authentifizierung und Autorisierung sowie der Test der Schnittstellen und Netzwerkverbindungen, und nicht zuletzt die WLAN-Integration der Raspberry Pi-Plattform.

Ausgeschlossen aus dem Projektumfang wurde die direkte Kommunikation mit 3D-Druckern (aufgrund fehlender Schnittstellen), die Integration in das unternehmensweite Intranet (Zeitrestriktionen), die Übertragung von Druckdaten oder (On-Device) Statusüberwachung der Drucker sowie umfangreiche Hardware-Modifikationen bestehender Geräte.

Die Integration in das unternehmensweite Intranet war ursprünglich eingeplant. Aufgrund zeitlicher Beschränkungen und der begrenzten Projektdauer musste dieser Aspekt jedoch zurückgestellt werden. Die notwendigen administrativen Prozesse innerhalb des Konzerns, insbesondere die Genehmigungsverfahren und die Beantragung der erforderlichen Zertifikate, hätten den vorgegebenen Zeitrahmen überschritten. Daher wurde in Abstimmung mit Herrn Noack und Kollegen entschieden, diese Integration zu einem späteren Zeitpunkt nach Projektabschluss zu realisieren.

#### 1.4 Projektumfeld und betriebliche Schnittstellen

Das Projekt wurde im Rahmen der Ausbildung zum Fachinformatiker für digitale Vernetzung in der TBA durchgeführt. Die organisatorischen Rahmenbedingungen sind somit durch konzerninternen Sicherheitsrichtlinien und zeitintensiven, schwer einsehbaren IT-Prozesse geprägt.

Die zentralen Schnittstellen umfassten die IT-Abteilung für die Genehmigung von Netzwerkkonfigurationen und netzwerktechnischen Gangbarkeiten, die Ausbildungsleitung (in dem Fall insbesondere meinen Ausbilder Martin Noack) für fachliche Betreuung und Ressourcenbereitstellung, die Endanwender - während des Projektes noch primär im Sinne der Ausbilder - ebenfalls jedoch in Form von Auszubildenden der TBA sowie die Hardware-Integration über die smarten Tapo-Steckdosen als agierende IoT-Gateways zu den 3D-Druckern.

## 2. Projektplanung

### 2.1 Zeitplanung nach V-Modell

Die Projektplanung folgte dem V-Modell mit agilen Elementen, unterteilt in fünf Projektphasen. Der Gesamtzeitraum erstreckte sich vom 15. April bis 05. Juni 2025 mit einem ursprünglich geplanten Projektumfang von 36 Stunden, der mit 38 Stunden geringfügig überschritten wurde.

Phase 1 (15.-19. April, 6 Stunden) Jene Phase widmete sich der Projektplanung und -Analyse. Die Analyse der aktuellen Prozesse und Systeme, die Bewertung der vorhandenen Netzwerkarchitektur sowie die Prüfung der Sicherheitsanforderungen standen dabei im Fokus. Regelmäßige Abstimmungen mit meinem Ausbilder halfen, die Planung zu präzisieren und aus den Erfahrungen mit den systeminternen Prozessen des ehemaligen Azubis zu lernen.

Phase 2 (22.-26. April, 6 Stunden) konzentrierte sich auf die Entwicklung der Systemarchitektur und Schnittstellenkonzeption. Die Definition der Kommunikationswege (WLAN, API), die Auswahl geeigneter Hard- und Softwarekomponenten sowie die Planung des konzeptionellen Aufbaus (Nutzer-



und Rechtverwaltung, Funktionalitäten und entsprechende API-Routen etc.) bildeten Schwerpunkte dieser Phase.

Phase 3 (29. April - 10. Mai, 16 Stunden) umfasste die Umsetzung. Diese Phase beinhaltete die Konfiguration des Raspberry Pi für die WLAN-Druckeranbindung, die Entwicklung des Webportals, den Aufbau der Datenbank sowie die Implementierung der Authentifizierung und Autorisierung. Die zwei Stunden Mehraufwand entstanden durch unerwartete Herausforderungen bei der Smart-Plug-Integration.

Phase 4 (13.-17. Mai, 6 Stunden) diente dem Test und der Optimierung. Funktionstest mit mehreren Nutzern und Druckern, Fehleranalyse und -behebung sowie die Optimierung der Datenverarbeitung und Darstellung standen im Mittelpunkt dieser Phase.

Phase 5 (20. Mai - 5. Juni, 4 Stunden) wurde für die Dokumentation genutzt. Die Beschreibung der Systemarchitektur und Abläufe, die Darstellung der technischen Umsetzung und die Bewertung der erreichten Ziele und möglicher Erweiterungen wurden in dieser Phase abgeschlossen.

## 2.2 Ressourcenplanung

Die Hardware-Komponenten umfassten einen Raspberry Pi 5 (8GB RAM, 128GB Speicher) als zentrale Serverplattform, nachdem sich der ursprünglich vorgesehene Raspberry Pi 4 als unterdimensioniert erwiesen hatte. Sechs TP-Link Tapo P110 Smart-Plugs bildeten das Herzstück der IoT-Integration. Ein 19-Zoll-Serverschrank wurde für die professionelle Unterbringung beschafft, ergänzt durch privat eingabachte Komponenten wie Lüftereinheiten und Kabelmanagement-Systeme.

Der Software-Stack basierte vollständig auf Open-Source-Technologien. Das Backend wurde mit Python 3.11, Flask 2.3, SQLAlchemy 2.0 und SQLite implementiert. Für das Frontend wurde nach dem Scheitern der Next.js-Integration eine eigene Lösung mit HTML/CSS/JavaScript als Notlösung entwickelt, wobei für die Interface-Entwicklung KI-Unterstützung verwendet wurde, da dies meiner Fachrichtung zu wider ist. Das System läuft auf Raspbian OS mit systemd-Services und OpenSSL. Die IoT-Gateway Integration zu den smarten Steckdosen der jeweiligen 3D-Drucker erfolgte über die PyP100-Bibliothek.

Die Gesamtkosten beliefen sich auf weniger als die Hälfte - sprich < 300 Euro, deutlich unter der ursprünglichen Kalkulation von 600 Euro. Die Kostenersparnis resultierte aus der Nutzung vorhandener Hardware-Komponenten und der breitläufig kompatiblen Umsetzbarkeit des Projektes.

## 2.3 Qualitätssicherungsplanung

Das Qualitätssicherungskonzept orientierte sich am V-Modell. Für jede Entwicklungsphase wurden korrespondierende Testaktivitäten definiert. Eine VirtualBox-basierte Entwicklungsumgebung ermöglichte Backend-Tests ohne Gefährdung der Produktivumgebung. Hardware-in-the-Loop-Tests mit echten WLAN-Steckdosen gleichen Modells gewährleisteten realitätsnahe Testbedingungen. Die separate Produktionsumgebung auf dem Raspberry Pi diente finalen Systemtests.



Die Teststrategien umfassten isolierte Tests kritischer Komponenten mit einer breitläufigen und fast gänzlichen Codeabdeckung. Sicherheitstests fokussierten sich auf grundlegende Angriffsvektoren wie Brute-Forcing, SQL-Injection oder Man-in-the-Middle Attacken; entsprechend wurden Rate-Limits für den API, verschlüsselte Verbindungen via HTTPS, User-Input-Bereinigung und ein zusätzlich abgesicherter Kiosk-Modus installiert - mit der Experimentierfreudigkeit der Azubis im Hinterkopf.

### 3. Analyse und Bewertung

#### 3.1 Bewertung der vorhandenen Systemarchitektur

Der Ist-Zustand präsentierte sich als fragmentierte Landschaft. Ein Frontend-Prototyp (Next.js) existierte ohne Backend-Anbindung, die 3D-Drucker operierten isoliert ohne Netzwerkfähigkeit, die Reservierungsverwaltung erfolgte analog über ein Whiteboard, und ein Raspberry Pi 4 stand als ungenutzter Server zur Verfügung.

Die identifizierten Defizite umfassten die fehlende cyberphysisch-vernetzte Integration, keine zentrale oder längerfristige Datenhaltung, ineffiziente manuelle Prozesse sowie Probleme bei der Zuweisung von Verantwortlichkeiten durch die analoge Verwaltung. Diese Analyse bildete die Grundlage für die Entwicklung einer integrierten Lösung.

#### 3.2 Bewertung der heterogenen IT-Landschaft

Die IT-Infrastruktur der TBA präsentierte sich als gewachsene Umgebung ohne mir direkt ersichtliche bestehende, gesonderte VLAN-Strukturen. Die 3D-Drucker verschiedener Hersteller erforderten eine zusammenbindende Abstraktionsebene.

Der dementsprechend gewählte Lösungsansatz über IoT-Integration mittels WLAN-Steckdosen ermöglichte eine universelle Steuerung unabhängig vom Druckermodell durch Abstraktion auf die Stromversorgungsebene. Diese Entscheidung basierte nicht zuletzt auf meiner privaten Erfahrung mit TAPO-Geräten, wobei sich die programmatische Ansteuerung als deutlich komplexer herausstellte als die private Nutzung.

#### 3.3 Analyse der IT-sicherheitsrelevanten Bedingungen

Die Sicherheitsanforderungen des Unternehmens diktierten maßgeblich die Systemarchitektur. Keine permanente Internetverbindung war zulässig geschweige denn möglich, ein isoliertes Netzwerksegment für IoT-Komponenten wurde erforderlich, selbstsignierte SSL-Zertifikate mussten mangels Möglichkeiten wie bspw. "Let's Encrypt" verwendet werden; Compliance - zumindest was das Intranet betraf - war obligatorisch.

Die implementierten Sicherheitsmaßnahmen umfassten neben den bereits genannten zusätzlich auch Passwort-Hashing mit bcrypt, CSRF-Schutz und Session-Management sowie Firewall-Regeln mit Port-Beschränkung. Die Verwendung von FirewallD erfolgte der Vertraulichkeit halber, da ich mit diesem System bereits umfangreiche Erfahrungen gesammelt hatte. Besonders wichtig war die Implementierung von HTTPS auch im Kiosk-Modus, um für den unwahrscheinlichen Fall eines Man-in-the-Middle-Angriffs auf dem Gerät selbst keine Klartextpasswörter im Netzwerkstrom zu haben –



eine Entscheidung, die wahrscheinlich mehr meinem Gewissen als der letztendlichen Eintrittswahrscheinlichkeit entsprach.

### 3.4 Anforderungsgerechte Auswahl der Übertragungssysteme

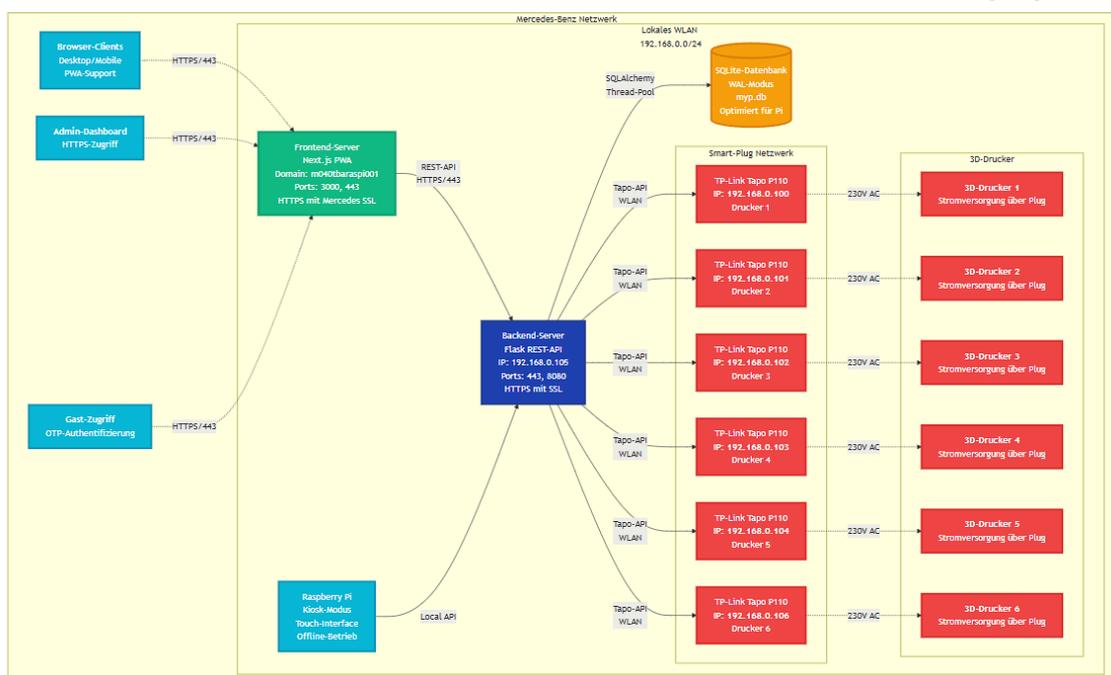
Die Evaluierung verschiedener Optionen ergab, dass eine direkte 3D-Drucker-Integration aufgrund fehlender Schnittstellen nicht möglich war. Cloud-basierte Lösungen schieden wegen der Offline-Anforderung aus. Als hinreichende Lösung wurde als die Fernsteuerung der Steckdosen identifiziert.

Die technische Herausforderung bestand darin, dass die TP-Link Tapo P110 über keine dokumentierte API verfügten (proprietär, verschlüsselt). Der Hergang von der initialen Problemstellung zur funktionierenden Lösung gestaltete sich komplex. Zunächst versuchte ich, ein recherchiertes Python-Modul für die Steuerung zu verwenden, was jedoch scheiterte. Daraufhin führte ich eine systematische Protokollanalyse mittels Wireshark durch. Die Untersuchung des Netzwerkverkehrs zwischen der TAPO-App und den Steckdosen offenbarte eine verschlüsselte Kommunikation mit dynamisch generierten Session-Keys. Nach mehreren Tagen intensiver Analyse und verschiedenen Implementierungsversuchen konnte ich schließlich PyP100 als funktionsfähige Python-Bibliothek identifizieren, die das proprietäre Kommunikationsprotokoll korrekt implementierte und eine lokale Steuerung ohne Cloud-Abhängigkeit oder Eigenlösung ermöglichte.

## 4. Systemarchitektur und Schnittstellenkonzeption

### 4.1 Gesamtsystemarchitektur

Die durch das Projekt entstandene Architektur umfasste nun also den Backend-Server, welcher zuweilen im Rahmen der Projektanpassung auch als Frontend-Server operiert. Dies hat den Vorteil, das auf einen Raspberry Pi verzichtet werden kann und ein solcher also der Ausbildung mehr zur Verfügung steht. Der Backend-Server steuert die an die 3D-Drucker angebotenen WLAN-Steckdosen an und stellt das Frontend im Kiosk-Modus auf dem TBA-lokalen Terminal zur Verfügung.





## 4.2 Technische Systemarchitektur

Das Schichtenmodell umfasst die Präsentationsschicht als Web-Frontend (HTTPS/Port 443), die Anwendungsschicht mit Flask-Backend und REST-API, die Geschäftslogikschicht für Reservierungsmanagement und Scheduler, die Datenhaltungsschicht mit SQLite-Datenbank, die IoT-Integrationsschicht für die Tapo-Steckdosen sowie die Hardwareschicht mit stromgesteuerten 3D-Druckern.

## 4.3 Schnittstellenkonzeption

Das REST-API-Design umfasst vielerlei Endpunkte, strukturiert in logische Bereiche. Die Authentifizierung erfolgt über `/api/auth/` mit Login, Logout und Session-Management. Die Benutzerverwaltung ist unter `/api/users/` für CRUD-Operationen implementiert. Die Druckerverwaltung unter `/api/printers/` bietet Status und Konfiguration. Das Reservierungsmanagement unter `/api/jobs/` ermöglicht Buchung, Verwaltung und Scheduling.

Die IoT-Schnittstelle kommuniziert über HTTP/TCP via WLAN mit session-basierter Authentifizierung und dynamischen Tokens. Die verfügbaren Operationen umfassen Power On/Off, Status-Abfrage und Energiemessung mit implementierter Fehlerbehandlung durch Retry-Mechanismen und Timeout-Handling; dies war insbesondere für den Kiosk Modus wichtig.

## 4.4 Datenmodell

Die Kernentitäten des Datenmodells umfassen User für Benutzerkonten mit Rollen und Berechtigungen, Printer für 3D-Drucker-Definitionen mit Steckdosen-Zuordnung, Job für Reservierungen mit Zeitfenstern und Status, SmartPlug für IoT-Geräte-Konfiguration und Zustandsverwaltung.

## 5. Umsetzung

### 5.1 Implementierung der Backend-Infrastruktur

Die Flask-Anwendungsstruktur folgt einer modularen Blueprint-Architektur.

Die zentrale Implementierungsherausforderung bestand in der Smart-Plug-Integration. Nach dem beschriebenen Hergang von Wireshark-Analyse zu PyP100 erwies sich diese Bibliothek als einzige funktionsfähige Lösung. Die Scheduler-Implementation erforderte sorgfältige Synchronisation:

```
1 class SmartPlugScheduler:
2     def __init__(self):
3         self.scheduler = BackgroundScheduler()
4         self.lock = threading.Lock()
5
6     def schedule_job(self, job_id, start_time, duration):
7         with self.lock:
8             self.scheduler.add_job(...)
```



## 5.2 IoT-Integration und Hardware-Steuerung

Die Smart-Plug-Konfiguration erfolgte mit statischen IP-Adressen im Bereich 192.168.0.100-106. Die lokale Authentifizierung funktioniert ohne Cloud-Service-Abhängigkeit.

Die Kommunikation wurde in einer Manager-Klasse gekapselt:

```
1 class SmartPlugManager:
2     def __init__(self, plug_configs):
3         self.plugs = {id: Tapo(ip, user, pass) for id, ip in plug_configs.items()}
4
5     async def control_printer(self, printer_id, action):
6         plug = self.plugs[printer_id]
7         return await plug.on() if action == 'start' else await plug.off()
```

## 5.3 Sicherheitsimplementierung

```
1 from flask_login import login_required, current_user
2 from werkzeug.security import generate_password_hash, check_password_hash
3 import secrets
4
5 class SecurityManager:
6     def __init__(self):
7         self.failed_attempts = {}
8         self.rate_limiter = RateLimiter(max_attempts=5, window=300)
9
10    def hash_password(self, password):
11        return generate_password_hash(password, method='bcrypt', salt_rounds=12)
12
13    def verify_login(self, username, password):
14        if self.rate_limiter.is_blocked(username):
15            raise SecurityException("Zu viele fehlgeschlagene Anmeldeversuche")
16
17        user = User.query.filter_by(username=username).first()
18        if user and check_password_hash(user.password_hash, password):
19            self.rate_limiter.reset(username)
20            return user
21
22        self.rate_limiter.record_failure(username)
23        return None
```

Die umfassende Sicherheitsarchitektur implementiert mehrschichtige Schutzmaßnahmen entsprechend den Unternehmensrichtlinien. Die Authentifizierung nutzt bcrypt (Cost-Factor 12) sichere Passwort-Speicherung. Das Session-Management über Flask-Login gewährleistet sichere Sitzungsverwaltung mit automatischem Timeout nach 30 Minuten Inaktivität.

Der CSRF-Schutz wurde für alle state-verändernden Operationen implementiert. Jede kritische Aktion erfordert ein gültiges CSRF-Token:

```
1 @app.before_request
2 def csrf_protect():
3     if request.method == "POST":
4         token = session.get('csrf_token', None)
5         if not token or token != request.form.get('csrf_token'):
6             abort(403)
7
8     def generate_csrf_token():
9         if 'csrf_token' not in session:
10            session['csrf_token'] = secrets.token_hex(16)
11            return session['csrf_token']
```



Das Rate-Limiting implementiert eine exponentielle Timeout-Strategie mit IP- und benutzerspezifischer Überwachung. Nach 5 fehlgeschlagenen Anmeldeversuchen wird der Account für exponentiell wachsende Zeiträume gesperrt (5, 15, 60 Minuten). Die Input-Validierung erfolgt mittels Python-Bibliothek "WTForms" mit automatischer XSS-Bereinigung für alle Benutzereingaben.

Zusätzliche Sicherheitsmaßnahmen umfassen HTTP-Security-Header, verschlüsselte Cookie-Übertragung mit Secure- und HttpOnly-Flags.

## 5.4 Systemkonfiguration und Deployment

Die Systemd-Service-Konfiguration gewährleistet automatischen Start und Überwachung:

```
1 [Unit]
2 Description=MYP HTTPS Backend Service
3 After=network.target
4
5 [Service]
6 Type=simple
7 User=myp
8 WorkingDirectory=/opt/myp
9 ExecStart=/usr/bin/python3 app.py --production
10 Restart=always
11 RestartSec=10
12
13 [Install]
14 WantedBy=multi-user.target
```

Das SSL-Zertifikat-Management erfolgt über selbstsignierte Zertifikate für den Offline-Betrieb. Die HTTPS-Implementierung auch im Kiosk-Modus gewährleistet verschlüsselte Passwortübertragung.

## 6. Test und Optimierung

### 6.1 Testdurchführung und-ergebnisse

Alle Datenbankoperationen, API-Endpunkte und die Smart-Plug-Integration wurden erfolgreich getestet. Die Integrationstests bestätigten die Funktionalität der produktiv eingesetzten Lösung und des Tapo-Steckdosen Controllings. Systemtests verifizierten Reservierungsszenarien mit bis zu 10 parallelen Sessions.

Die Performance-Optimierungen umfassten das Hardware-Upgrade von Raspberry Pi 4 auf Pi 5 aufgrund unzureichender Leistung, Datenbankindizierung für häufige Abfragen sowie Caching-Strategien für Smart-Plug-Status.

### 6.2 Sicherheitstests

Die Sicherheitstests fokussierten sich auf die Unternehmensrichtlinien und umfassten Netzwerkskans der IT. Zudem aktivierte sich das Rate-Limiting nach 5 fehlgeschlagenen Login-Versuchen erfolgreich in manuellen Tests.



## 6.3 Systemstabilität und Monitoring

Das implementierte Monitoring nutzt strukturiertes Logging mit Rotation:

```
1 import logging
2 from logging.handlers import RotatingFileHandler
3
4 logging.basicConfig(
5     handlers=[RotatingFileHandler('app.log', maxBytes=1000000, backupCount=10)],
6     level=logging.INFO,
7     format='%(asctime)s %(levelname)s %(name)s %(message)s'
8 )
```

## 7. Projektabschluss

### 7.1 Soll-Ist-Vergleich

Vollständig erreichte Ziele umfassen die webbasierte Reservierungsplattform, automatische Hardware-Steuerung via IoT, zentrale Verwaltungsoberfläche, robuste Authentifizierung und Rechteverwaltung, WLAN-Integration der Raspberry Pi-Plattform, Datenbankaufbau für Reservierungsverwaltung sowie Test der Schnittstellen und Netzwerkverbindungen.

Zusätzlich realisierte Features beinhalten Nutzungsstatistiken und Dashboard, erweiterte Sicherheitsfeatures wie Rate-Limiting und CSRF-Schutz sowie einen Kiosk-Modus für Werkstatt-Terminals.

Abweichungen vom ursprünglichen Plan waren die Konsolidierung auf einen statt zwei Raspberry Pis aus Kostengründen, das Hardware-Upgrade Pi 4 auf Pi 5 wegen Performance-Anforderungen, die Implementierung einer eigenen Frontend-Lösung als Notlösung statt Next.js-Integration sowie die Verschiebung der Intranet-Integration aufgrund von Zeitrestriktionen. Der Zeitverzug konnte durch effiziente Arbeitsweise in den finalen Sprints teilweise kompensiert werden; nicht jedoch vollständig.

### 7.2 Projektergebnisse und Erkenntnisse

Das MYP-System transformiert erfolgreich die analoge 3D-Drucker-Verwaltung in ein modernes System. Die Lösung demonstriert, dass auch Legacy-Hardware in moderne Systemlandschaften integriert werden kann.

Die zentralen Erfolgsfaktoren waren die pragmatische Abstraktion komplexer Hardware-Probleme, eine robuste Softwarearchitektur mit umfassender Fehlerbehandlung sowie die konsequente Berücksichtigung von Sicherheitsanforderungen von Projektbeginn an.

Als wichtige Erkenntnisse kristallisierten sich heraus, dass Hardware-Kompatibilitätsprüfungen vor Projektstart essentiell sind, Backup-Strategien für kritische Konfigurationen unerlässlich sind und agile Anpassungsfähigkeit bei unvorhergesehenen Problemen den Projekterfolg sichert. Zudem nehme ich mir persönlich mit, dass ich - auch wenn ich von Herzen dabei bin - eigenen Ehrgeiz einem pünktlichen Projektabschluss zu Gunsten zurückstellen muss.



### 7.3 Optimierungsmöglichkeiten

Das MYP-System bietet eine solide Basis für zukünftige Erweiterungen. Die modulare Architektur und umfassende API ermöglichen die Integration zusätzlicher Funktionalitäten ohne grundlegende Systemänderungen.

Kurzfristig ist die Anbindung an das unternehmenseigene Netzwerk geplant. Mittelfristig könnte bei Verfügbarkeit modernerer 3D-Drucker eine direkte Geräteintegration realisiert werden. Langfristig bietet sich die Erweiterung zu einer umfassenden Maker-Space-Management-Lösung an, die auch andere Gerätetypen wie Lasercutter oder CNC-Fräsen einbindet.

### 7.4 Formale Projektabschluss

Die Erstabnahme erfolgte am 3. Juni 2025 durch die Ausbildungsleitung der TBA. Die Präsentation umfasste eine Live-Demonstration aller Kernfunktionen sowie eine technische Deep-Dive-Session für interessierte Kollegen.

Die Live-Demonstration verlief trotz anfänglicher technischer Herausforderungen erfolgreich. Aktuell befindet sich das System in aktiver Beobachtung der realen Anwendung. Als letzter Schritt verbleibt die Schulung der Mitarbeiter, die in den kommenden Wochen stattfinden wird.



## Glossar

<b>Begriff</b>	<b>Erläuterung</b>
bcrypt	Kryptographische Hash-Funktion für sichere Passwort-Speicherung mit konfigurierbarem Cost-Faktor
CSRF	Cross-Site Request Forgery - Angriffsmethode, bei der authentifizierte Nutzer unbeabsichtigt Aktionen ausführen
Flask	Python-basiertes Web-Framework für Backend-Entwicklung. Grundgerüst des MYP-Servers mit REST-API und Session-Management
Hardware in the Loop	Testmethode, bei der reale Hardware-Komponenten in die Testumgebung eingebunden werden
Next.js	React-basiertes Frontend-Framework, ursprünglich von Torben Haack für den Prototyp verwendet
Scheduler	Zeitgesteuerte Komponente für automatisierte Smart-Plug-Operationen und Reservierungsverwaltung
Threads	Parallele Ausführungsstränge innerhalb eines Prozesses für gleichzeitige Operationen



## Anlagen

### Übergabeprotokoll

Mercedes-Benz AG - Technische Berufsausbildungsstätte  
03.06.2025

## Übergabeprotokoll - MYP System

**Datum:** 03.06.2025**Projekt:** MYP - Manage Your Printer

### Übergebende Partei

Name: Till Tomczak

Funktion: Auszubildender, Projektleiter

### Übernehmende Partei

Name: Martin Noack

Funktion: Ausbilder, Technische Berufsausbildungsstätte

### Übergebene Komponenten

#### Hardware

Komponente	Anzahl	Zustand
Raspberry Pi 5 (8GB)	1	Neuwertig, funktionsfähig
TP-Link Tapo P110	1	Neuwertig, konfiguriert
19-Zoll Serverschrank	1	Neuwertig, aufgebaut
Netzwerkkabel Cat6	3	Neuwertig
Netzteil Raspberry Pi	1	Neuwertig
Netzwerk Switch	1	Neuwertig



Mercedes-Benz AG - Technische Berufsausbildungsstätte  
03.06.2025

## Software

- MYP Backend (Flask-Applikation) v3.0
- SQLite Datenbank mit Initialstruktur
- Konfigurationsdaten

## Dokumentation

- Technische Dokumentation
- Benutzerhandbuch
- API-Dokumentation
- Administratorhandbuch
- Security Scan Report
- Zugangsdaten

Administrator-Account: admin@mercedes-benz.com (Passwort separat übergeben)

## Systemstatus

- System vollständig installiert und konfiguriert
- Security Scan bestanden (Rolf Christiansen)
- Vorheriges Frontend verworfen
- Netzwerkanbindung fehlgeschlagen
- Alle funktionalen Tests erfolgreich durchgeführt
- Dokumentation vollständig
- Schulungsmaterial erstellt, Schulung geplant

## Bemerkungen

Trotz erheblicher technischer Herausforderungen - einschließlich der Notwendigkeit einer kompletten Frontend-Neuentwicklung und ungelöster Netzwerkintegration - gelang es dem Auszubildenden, ein funktionsfähiges System zu entwickeln.

Till Tomczak (Auszubildender)

Martin Noack (Auftraggeber)



## Screenshots des Projektarbeitsprozesses

tp-link
TP-Link Wireless N Nano Router WR802N
Model No. TL-WR802N

- Status
- Quick Setup
- Operation Mode
- Network
- Wireless
- Guest Network
- DHCP
- DHCP Settings
- DHCP Clients List
- Address Reservation
- System Tools
- Logout

### DHCP Clients List

This page displays information of all DHCP clients on the network.

ID	Client Name	MAC Address	Assigned IP	Lease Time
1	P115	98:25:4A:E1:2D:3C	192.168.0.101	Permanent
2	P115	98:25:4A:E1:35:FE	192.168.0.100	Permanent
3	P115	98:25:4A:E1:A3:98	192.168.0.103	Permanent
4	P115	98:25:4A:E1:2D:A6	192.168.0.104	Permanent
5	P115	E4:FA:C4:EB:4D:08	192.168.0.102	Permanent
6	P115	98:25:4A:E1:1D:8E	192.168.0.106	Permanent
7	C040L0079726760	08:8E:90:E6:E5:88	192.168.0.108	01:58:07

Anlage 1: Statische IP-Konfiguration auf dem WLAN-Router des Offline-TBA-Netzwerkes

tp-link
TP-Link Wireless N Nano Router WR802N
Model No. TL-WR802N

### Status

---

Firmware Version: 0.9.1.3.17 v0001.0 Build 200623 Rel 63672n  
Hardware Version: TL-WR802N v4 00000004

---

### LAN

MAC Address: 9C:A2:F4:63:70:56  
IP Address: 192.168.0.1  
Subnet Mask: 255.255.255.0

---

### Wireless 2.4GHz

Operation Mode: **Access Point**  
Wireless Radio: Enabled  
Name(SSID): TP-Link\_7056  
Mode: 11bgn mixed  
Channel: Auto(Channel 2)  
Channel Width: Auto  
MAC Address: 9C:A2:F4:63:70:56

Anlage 2: WLAN-Router Konfiguration im Access-Point Modus zur Sicherstellung der Client Konnektivitäten



## API-Dokumentation

- ▼ MYP Plattform - Systeme...
- Projektübersicht
- ▼ Architektur-Übersicht
  - Core-Struktur
  - ▼ Design-Patterns
    1. Datenbank-Se...
    2. Berechtigungs...
    3. Konflikt-Mana...
    4. Logging-Strat...
- ▼ Installation und Setup
  - Systemvoraussetz...
  - ▼ Installationsmodi
    1. Abhängigkeite...
    2. Vollständige Ki...
  - ▼ Systemd-Services
    - myp-https.service
    - myp-kiosk.service
    - kiosk-watchdog...
    - myp-firewall.serv...
  - Verzeichnisstruktur
- ▼ Konfiguration
  - SSL-Zertifikate
  - Netzwerk-Sicherheit
  - Remote-Zugang
- ▼ Performance-Optimi...
- Raspberry Pi-spezif...
- ▼ Integration und Sch...
- TP-Link Tapo Smar...
- E-Mail-Benachrichtl...
- Datei-Uploads

### H1 MYP Plattform - Systemdokumentation

#### H2 Projektübersicht

MYP (Manage Your Printers) ist ein umfassendes 3D-Drucker-Verwaltungssystem für Mercedes-Benz, das für den Betrieb auf Debian/Linux-Systemen (speziell Raspberry Pi) im HTTPS-Kiosk-Modus entwickelt wurde. Das System verwaltet Drucker-Terminplanung, Benutzerauthentifizierung, Job-Warteschlangen und Smart-Plug-Integration mit TP-Link Tapo-Geräten.

#### H2 Architektur-Übersicht

### H3 Core-Struktur

Die Anwendung folgt einer Flask-Blueprint-Architektur mit klarer Trennung der Zuständigkeiten:

- **app.py**: Hauptanwendungs-Einstiegspunkt mit HTTPS-Konfiguration und Raspberry Pi-Optimierungen
- **models.py**: SQLAlchemy-Modelle mit SQLite-Optimierung (WAL-Modus, Caching, Performance-Tuning)
- **blueprints**: Modulare Feature-Implementierungen
  - **auth.py**: Authentifizierung und Session-Management
  - **admin.py & admin\_api.py**: Administrative Schnittstellen und APIs
  - **printers.py**: Drucker-Verwaltung und Smart-Plug-Integration
  - **jobs.py**: Job-Warteschlangen-Management
  - **guest.py**: Gastzugang mit OTP-System
  - **calendar.py**: Terminplanung mit Konflikt-Management
  - **users.py & user.py**: Benutzerverwaltung und Profile

#### H3 Design-Patterns

#### H4 1. Datenbank-Sessions

Verwendet Scoped Sessions mit ordnungsgemäßer Bereinigung:

```

</> 1 with get_db_session() as session:
```

Anlage 3: API-Dokumentation (.md / Markdown Format) des Projektes

## Benutzerhandbuch

- ▼ MYP Plattform - Benutz...
- Inhaltsverzeichnis
- ▼ Übersicht
  - Hauptfunktionen
- ▼ Admin-Funktionen
  - ▼ Zugriff auf das Ad...
  - Benutzer-Verwal...
  - Drucker-Manage...
  - Job-Verwaltung
  - System-Administ...
  - ▼ Gastanfragen-Ver...
  - Genehmigungsw...
  - Ablehnungswork...
  - Benutzeroberflä...
- ▼ Gastauftrag-System...
- Übersicht
- OTP-Generierung u...
- Status-Abfrage
- Status-Informationen
- Benutzer-Workflow ...
- ▼ Warteschlangen-Sy...
- Übersicht
- Universelle Drucke...
- Intelligente Job-Ers...
- Background-Überw...
- Benutzer-Workflow ...
- Benachrichtigungs...
- ▼ Benutzerinterface
  - Frontend-Architektur
  - Glassmorphism UI...

### H1 MYP Plattform - Benutzerhandbuch

#### H2 Inhaltsverzeichnis

1. [Übersicht](#)
2. [Admin-Funktionen](#)
3. [Gastauftrag-System mit OTP](#)
4. [Warteschlangen-System](#)
5. [Benutzerinterface](#)
6. [Drucker-Management](#)
7. [Job-Verwaltung](#)
8. [Sicherheit und Berechtigungen](#)

#### H2 Übersicht

Die MYP Plattform ist ein umfassendes 3D-Drucker-Verwaltungssystem, das speziell für Mercedes-Benz entwickelt wurde. Es ermöglicht die effiziente Verwaltung von 3D-Druckaufträgen, Benutzern und Druckern in einer sicheren, webbasierten Umgebung.

#### H3 Hauptfunktionen

- **Benutzer- und Rollenverwaltung** mit granularen Berechtigungen
- **3D-Drucker-Management** mit Smart-Plug-Integration
- **Job-Planung und -Überwachung** mit Konfliktmanagement
- **Gastauftrag-System** mit sicherem OTP-Code-System
- **Warteschlangen-Management** für Offline-Drucker
- **Admin-Dashboard** mit umfassenden Verwaltungsfunktionen

#### H2 Admin-Funktionen

#### H3 Zugriff auf das Admin-Dashboard

Das Admin Dashboard ist unter `/admin` verfügbar und erfordert Admin Berechtigung.

Anlage 4: Benutzeranleitung (.md / Markdown Format) für das Projekt



Vernetzte 3D-Druck-Reservierungsplattform mit IoT-Anbindung und zentraler Verwaltungsoberfläche

(geplante) Schulungspräsentation

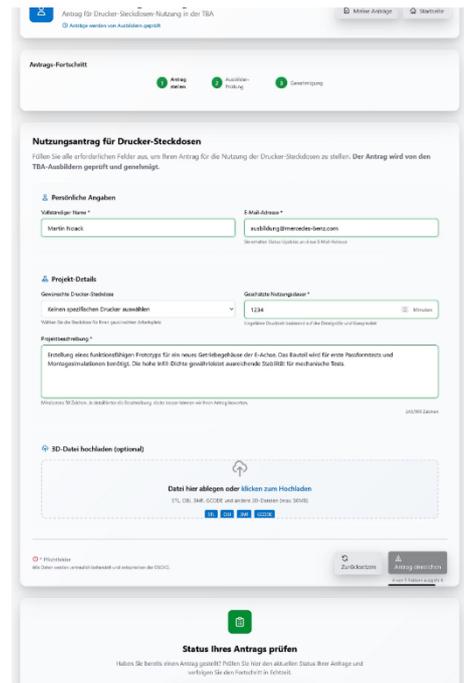


# Mercedes-Benz - Manage Your Printer System

Das Mercedes-Benz Manage Your Printer (MYP) System ermöglicht die Verwaltung von 3D-Druckaufträgen für die Ausbildungswerkstatt. Diese Präsentation erklärt den Antragsprozess für Gastnutzer.

## Antragsformular ausfüllen

- 
**TBA-Nutzungsantrag**  
 Technisches Betriebsmittel Antrag für Drucker-Steckdosen.
- 
**Zeitangabe**  
 Gewünschte Nutzungsdauer in Minuten ist ein Pflichtfeld.
- 
**Dateien (optional)**  
 Maximale Dateigröße beachten beim Hochladen.



Anlage 5 (2 Stück): Geplante Folien zur Schulung der Mitarbeiter (Ausschnitt)



## Testprotokolle

```

361 # ergebnisse zusammenfassen
362 print("=" * 70)
363 print("📊 TESTERGEBNISSE ZUSAMMENFASSUNG")
364 print("=" * 70)
365
366 success_count = 0
367 warning_count = 0
368 failed_count = 0
369 skipped_count = 0
370
371 for test_name, result in all_results.items():
372     status = result["status"]
373     message = result["message"]
374
375     if status == "SUCCESS":
376         print(f"✅ {test_name}: {message}")
377         success_count += 1
378     elif status == "WARNING":
379         print(f"⚠️ {test_name}: {message}")
380         warning_count += 1
381     elif status == "FAILED":
382         print(f"❌ {test_name}: {message}")
383         failed_count += 1
384     elif status == "SKIPPED":
385         print(f"⏸️ {test_name}: {message}")
386         skipped_count += 1
387
388 total_tests = len(all_results)
389
390 print("\n" + "=" * 70)
391 print("📊 STATISTIKEN")
392 print("=" * 70)
393 print(f"Gesamt: {total_tests} Tests")
394 print(f"✅ Erfolgreich: {success_count}")
395 print(f"⚠️ Warnungen: {warning_count}")
396 print(f"❌ Fehlgeschlagen: {failed_count}")

```

Test-ID	Komponente	Prüfung	Dauer
T001	Syntax	Python-Kompilierung beider App-Versionen	~0.5s
T002	Import	Erfolgreiche Modul-Imports	~1.2s
T003	Models	Datenbank-Modell-Validierung	~1.4s
T004	Blueprints	Route-Architektur-Tests	~1.3s
T005	Flask-App	App-Objekt-Erstellung	~1.1s
T006	Dependencies	Version-Kompatibilität	~1.5s
T007	Code-Metriken	Zeilen-Reduktion, Komplexität	~0.1s

### Anlage 6: Ausschnitt der automatisiertes Testprotokolle



## Screenshots der Benutzeroberfläche

## TBA-Nutzungsantrag

Antrag für Drucker-Steckdosen-Nutzung in der TBA

[Anträge werden von Ausbildern geprüft](#)

[Meine Anträge](#) [Startseite](#)

### Antrags-Fortschritt

- 1 Antrag stellen
- 2 Ausbilder-Prüfung
- 3 Genehmigung

### Nutzungsantrag für Drucker-Steckdosen

Füllen Sie alle erforderlichen Felder aus, um Ihren Antrag für die Nutzung der Drucker-Steckdosen zu stellen. **Der Antrag wird von den TBA-Ausbildern geprüft und genehmigt.**

#### Persönliche Angaben

Vollständiger Name \*

E-Mail-Adresse \*

Sie erhalten Status-Updates an diese E-Mail-Adresse

#### Projekt-Details

Gewünschte Drucker-Steckdose

Geschätzte Nutzungsdauer \*  Minuten

Wählen Sie die Steckdose für Ihren gewünschten Arbeitsplatz Ungefähre Druckzeit basierend auf der Dateigröße und Komplexität

Projektbeschreibung \*

Erstellung eines funktionsfähigen Prototyps für ein neues Getriebegehäuse der E-Achse. Das Bauteil wird für erste Passformtests und Montagesimulationen benötigt. Die hohe Infill-Dichte gewährleistet ausreichende Stabilität für mechanische Tests.

Mindestens 50 Zeichen. Je detaillierter die Beschreibung, desto besser können wir Ihren Antrag bewerten. 245/500 Zeichen

#### 3D-Datei hochladen (optional)

**Datei hier ablegen oder klicken zum Hochladen**

STL, OBJ, 3MF, GCODE und andere 3D-Dateien (max. 50MB)

[STL](#) [OBJ](#) [3MF](#) [GCODE](#)

\* Pflichtfelder  
Alle Daten werden vertraulich behandelt und entsprechen der DSGVO.

[Zurücksetzen](#) [Antrag einreichen](#)

4 von 5 Feldern ausgefüllt

### Status Ihres Antrags prüfen

Haben Sie bereits einen Antrag gestellt? Prüfen Sie hier den aktuellen Status Ihrer Anfrage und verfolgen Sie den Fortschritt in Echtzeit.

[Meine Anträge](#) [Status abfragen](#)

Anlage 7, 8, 9: Screenshot der Benutzeroberflächen



**Mercedes-Benz** Manage Your Printer

Dashboard Drucker-Steckdosen Reservierungen Statistiken Schichtplan Anträge stellen Meine Anträge Ausbilder-Bereich

### Anfrage-Status

Verfolgen Sie den Status Ihrer Gastanfrage für 3D-Druck

Wird geprüft

#### Anfrage Details

Übersicht Ihrer eingereichten Gastanfrage

Anfrage-ID	Erstellt am
<b>#2</b>	<b>04.06.2025 14:32</b>
Name	Gewünschte Dauer
<b>Martin Noack</b>	<b>1234 Minuten</b>

Begründung  
Erstellung eines funktionsfähigen Prototyps für ein neues Getriebegehäuse der E-Achse. Das Bauteil wird für erste Passformtests und Montagesimulationen benötigt. Die hohe Infill-Dichte gewährleistet ausreichende Stabilität für mechanische Tests.

**Ihre Anfrage wird geprüft**  
Unser Team prüft Ihre Anfrage mit höchster Priorität. Sie erhalten eine sofortige Benachrichtigung, sobald eine Entscheidung getroffen wurde. Diese Seite aktualisiert sich automatisch alle 30 Sekunden.

### Meine Druckanträge

Übersicht Ihrer eingereichten Anträge für ausbildung@mercedes-benz.com

Neuen Antrag stellen

Aktualisieren Alle Anträge

GESAMT	PRÜFUNG	GENEHMIGT	ABGELEHNT
1	1	0	0

**#2** Wird geprüft

Antragsteller	E-Mail	Drucker	Dauer	
Martin Noack	ausbildung@mercedes-benz.com	Automatisch zuweisen	1234 Min	04.06.2025 14:32 Uhr

**In Bearbeitung**

Projektbeschreibung  
Erstellung eines funktionsfähigen Prototyps für ein neues Getriebegehäuse der E-Achse. Das Bauteil wird für erste Passformtests und Montagesimulationen benötigt. Die hohe Infill-Dichte gewährleistet ausreichende Stabilität für mechanische Tests.

**Mercedes-Benz**  
TBA Marienfelde - Steckdosen-Steuerung

Das Beste oder nichts - Professionelles 3D-Druck Management für Mercedes-Benz.

#### System

Version: 3.0.0  
Status: ● Online

Steckdosenschaltzeiten

#### Benachrichtigungen

Nicht stören: Inaktiv

Alle Benachrichtigungen aktiv

#### Rechtliches



Vernetzte 3D-Druck-Reservierungsplattform  
mit IoT-Anbindung und zentraler Verwaltungsoberfläche

---