Sichere und effektive Nutzung von Technologiedatenmarktplätzen am Beispiel der Lasermarkierung

Vom Fachbereich Maschinenbau

an der Technischen Universität Darmstadt

zur Erlangung des akademischen Grades eines Doktor-Ingenieurs (Dr.-Ing.)

genehmigte

DISSERTATION

vorgelegt von

Ghaidaa Othman, M.Sc.

aus Latakia, Syrien

Berichterstatter:	Prof. Dr.-Ing. Reiner Anderl
Mitberichterstatter:	Prof. Dr.-Ing. Matthias Weigold
Tag der Einreichung:	04.12.2023
Tag der mündlichen Prüfung:	11.06.2024

Darmstadt 2024

Ghaidaa Othman: Technologiedatenmarktplätzen am Beispiel der Lasermarkierung
Darmstadt, Technische Universität Darmstadt

Jahr der Veröffentlichung der Dissertation auf TUprints: 2024

Tag der mündlichen Prüfung: 11.06.2024

INHALTSVERZEICHNIS

ABBILDUNGSVERZEICHNIS

TABELLENVERZEICHNIS

PROZEDURVERZEICHNIS

ABKÜRZUNGENVERZEICHNIS

IT	Informationstechnologie
E	Electronic
IKT	Informations- und Kommunikationstechnologie
STL	Standard Transformation Language
B2B	Business-to-Business
B2C	Business-to-Customer
C2B	Customer-to-Business
C2C	Customer-to-Customer
KI	Künstliche Intelligenz
CVIM	Common Vehicle Information Model
P2P	Peer-to-Peer
IIoT	Industrial Internet of Things
Laser	Light amplification by stimulated emission of radiation
RP	Rapid Prototyping
CAD	Computer Aided Design
CAM	Computer Aided Manufacturing
SLS	Selective laser sintering

NC	Numerical Control
CNC	Computerized Numerical Control
TD	Technologiedaten
TDMP	Technologiedatenmarktplatz
OT	Operation Technology
NIST	National Institute of Standards and Technology
BSI	Bundesamt für Sicherheit in der Informationstechnik
EULA	End User License Agreement
UML	Unified Modeling Language
OMG	Object Management Group
ERP	Enterprise-Resource-Planning
E2EE	End-to-End Encryption
AES	Advanced Encryption Standard
RSA	Rivest-Shamir-Adleman
ECDSA	Elliptic Curve Digital Signature Algorithm
ENISA	European Network and Information Security Agency
PKI	Public Key Infrastructure
IV	Initialisierungsvektor

2FA	Zwei-Faktor-Authentifizierung
URI	Uniform Resource Identifier
TLS	Transport Layer Security
API	Application Programming Interface
TPM	Trusted Platform Module
HTTP	Hypertext Transfer Protocol
JSON	JavaScript Object Notation
MSP	Membership Service Provider
OU	Organizational Units
CA	Certificate Authority
gRPC	Google Remote Procedure Call
REST	Representational State Transfer
CSV	Comma separated values
DDoS	Distributed Denial of Service
Usw.	und so weiter
Etc.	et cetera
bzw.	Beziehungswei

1 EINLEITUNG

In der vorliegenden Dissertation wird ein Konzept vorgestellt, das Anwendern bei der effektiven und sicheren Nutzung von Technologiedatenmarktplätzen im Bereich der Lasermarkierung assistiert. Das Konzept liefert Handlungsempfehlungen für die Auswahl von Lasermaschinen und Lizenzmodellen sowie für die effektive und sichere Bereitstellung von Technologiedaten für einen bestimmten Auftrag. Außerdem ermöglicht die entwickelte Lösung eine sichere Einbindung von Lasermaschinen in Technologiedatenmarktplätze, die lizenzierte Technologiedaten anbieten, und stellt sicher, dass Technologiedaten sowie zugehörige Lizenzinformationen geschützt sind. Der Ansatz ist dabei bedarfsorientiert und berücksichtigt spezifische Anwenderbedürfnisse sowie Kriterien wie die Beschaffenheit des Auftrags sowie die dafür verwendeten Verfahren, Technologien und Materialien. Das Ziel besteht darin, den Anwender methodenbasiert bei der Auswahl und Konfiguration einer Laserbearbeitungsmaschine für neue Fertigungsprozesse zu unterstützen. Hierbei werden wirtschaftlich effiziente Laserbearbeitungsprozesse sowie Mechanismen für die sichere Nutzung und Nachverfolgung erworbenen Technologiewissens bereitgestellt. Um dies zu erreichen, wird ein rechnergestütztes Werkzeug für die sichere und effektive Bereitstellung und Verteilung von Technologiedaten entwickelt.

Für Fertigungsprozesse mit Lasertechnologien bedarf es einer speziellen Maschineneinstellung, für die auf Erfahrungswissen der Anwender bzw. Hersteller zurückgegriffen wird. Dabei müssen zahlreiche Steuergrößen berücksichtigt werden wie zum Beispiel der Maschinentyp, die Lasertechnologie, die Prozessparameter und die Eigenschaften des Werkstücks. In der Laserfertigung führt die manuelle Einstellung von Laserbearbeitungsmaschinen durch den Anwender zu einem höheren Arbeitsaufwand. Durch die vielen Versuche kommt es zum einem massiven Material- und Zeitverlust, bis das optimale Prozessergebnis vorliegt. Die Nachbestellung solcher prozessspezifischen Technologiedaten vom jeweiligen Maschinenhersteller ist mit höheren Kosten und langen Lieferzeiten verbunden. In beiden Fällen sind die Kosten umso höher, je individueller die Laserfertigungsaufträge sind. Vor diesem Hintergrund ist das Erwerben von Technologiedaten bei sogenannten Technologiedatenmarktplätzen für die Laserbearbeitung sinnvoll. Diese Technologiedaten können dann von Anwendern nach Bedarf erworben, entsprechend lizenziert und unmittelbar zur Produktion geschickt werden. Wenn der Anwender mehrere Typen von Laserbearbeitungsmaschinen von unterschiedlichen Herstellern in Verwendung hat, steigt der

Arbeitsaufwand zur Bereitstellung und Verwaltung der erworbenen Technologiedaten und der zugehörigen Lizenzinformationen enorm. Da heute keine Ansätze existieren, die die Teilnahme an Technologiedatenmarktplätzen sicher gestalten, ohne die Echtzeitanforderungen in der Laserproduktion zu beeinflussen, entstehen Einschränkungen bezüglich der Effizienz von Laserbearbeitungsprozessen. Darüber hinaus existieren keine Ansätze, die den Anwender bei der effektiven Bereitstellung von Technologiedaten auf den Technologiedatenmarktplätzen unterstützen.

Um diesen Herausforderungen zu begegnen, wird im Rahmen dieser Dissertation ein Konzept zur Unterstützung des Anwenders bei der sicheren Nutzung von Technologiedatenmarktplätzen speziell im Bereich der Anwendungsfall entwickelt. Das Konzept besteht aus drei wichtigen Elementen. Das erste Element stellt Handlungsempfehlungen zur auftragsspezifischen Auswahl von geeigneten Laserbearbeitungsmaschinen bereit. In der Planungsphase eines Laserauftrags ermöglicht das Konzept dem Anwender, auf Grundlage von Daten eine geeignete Laserbearbeitungsmaschine auszuwählen. Dafür wurde ein systematisches Vorgehen zur Erfassung und Evaluierung der existierenden Laserbearbeitungsmaschinen konzipiert. Das zweite Systemelement dient der effizienten Auswahl von passenden Lizenzmodellen für einen bestimmten Auftrag. Hierfür wurde ein wissensbasiertes Assistenzsystem entwickelt, das den Anwender bei der Spezifizierung der passenden Lizenzmodelle basierend auf festgelegten Wissensregeln methodisch unterstützt und nicht zuletzt fokussiert das Konzept auf die Sicherheit und Nachverfolgbarkeit von Technologiedaten und der zugehörigen Lizenzinformationen. Hierbei wird ein Lizenzierungskonzept entwickelt, das die Möglichkeit bietet, Technologiedaten von verschiedenen Marktplätzen bedarfsgerecht und sicher zu erwerben und zu nutzen. Durch die Verwendung moderner Technologien wie Blockchain und intelligente Verträge in diesem Konzept wird die Nutzung von Technologiedaten im Einklang mit den Lizenzbedingungen überwacht.

Dieses Kapitel bietet einen Überblick über die in dieser Dissertation behandelten Forschungsarbeiten und die daraus abgeleiteten Forschungsfragen. Im weiteren Verlauf der Einleitung werden in Unterkapitel 1.1 die aktuelle Problemstellung sowie die Motivation für die Auswahl des Themenbereichs dargelegt. Auf Basis der beschriebenen Problemstellung werden in Unterkapitel 1.2 die Forschungsziele formuliert, die zur Lösung der vorgestellten Herausforderungen beitragen sollen. Schließlich präsentiert Unterkapitel 1.3 die Struktur der vorliegenden Dissertation.

1.1 Motivation und Problemstellung

Die rasante Fortentwicklung der Lasertechnologie bietet der Fertigungsindustrie erhebliche Vorteile wie Flexibilität, Qualitätssteigerung und Zeiteffizienz. Außerdem bietet Lasertechnologie immer wieder die Möglichkeit, neue und innovative Produkte zu fertigen [1].

Angesichts der Transition zur Industrie 4.0 und der sich daraus ergebenden Vernetzung aller industriellen Systeme eröffnen sich neue Perspektiven für innovative Geschäftsmodelle und Wachstumschancen [2], [3]. Mit der Digitalisierung spielen Geschäftsplattformen in der Industrie für neue Wertschöpfungsmöglichkeiten und für ein besseres Verhältnis zu den Kunden eine bedeutsame Rolle [4]. Die Datensammlung, -aufbereitung und -auswertung wird in den künftigen Produkten eine bedeutsame Rolle spielen und stellt ein neues Investitionsgut für Mehrwertschaffung im Unternehmen dar [5]. In der modernen Wirtschaft ist die Generierung und Verbreitung sowie die effektive Nutzung von Informationen zentral – deshalb wird auch von der "Informationsökonomie" gesprochen. Dieser Wandel wird vor allem durch die fortschreitende Digitalisierung vorangetrieben, die es ermöglicht, dass nicht nur physische, sondern zunehmend auch immaterielle Produkte produziert und vermarktet werden können. Dadurch verschiebt sich der Fokus von materiellen Werten hin zu Wissensgütern und digitalen Daten [6]. Die Digitalisierung spielt eine Schlüsselrolle in dieser Transformation und fördert den Verkauf von Informationsprodukten mit geringen Herstellungskosten und großen Gewinnspannen Zum Beispiel können Online-Marktplätze für digitale Produkte einfach entwickelt und schnell in Betrieb genommen werden [7]. Insbesondere die Rolle von Technologiedatenmarktplätzen als Plattformen für spezifische Fertigungsdaten tritt immer stärker in den Vordergrund. Sie ermöglichen eine flexible und kostengünstige Anpassung von Laserbearbeitungsprozessen an individuelle Kundenanforderungen. Die Digitalisierung birgt sowohl viele neue Chancen als auch Risiken, insbesondere im Hinblick auf die Datensicherheit [8], [9].

Diese Dissertation widmet sich der sicheren Nutzung eines zukunftsorientierten und auf den Bedarf des Kunden abgestimmten Geschäftsmodells des Technologiedatenmarktplatzes für die Lasermarkierung. Dieser Online-Marktplatz ermöglicht den Handel mit speziellen Fertigungsdaten, die von den Unternehmen bereits entwickelt wurden und daher ohnehin vorhanden sind. Inspiriert von existierenden digitalen Handelsmarkplätzen bietet der Technologiedatenmarktplatz eine Auswahl an einstellbaren Parameterwerten für Laserbearbeitungsmaschinen wie Laserleistung, Fokus und Geschwindigkeit. Diese Daten

basieren auf dem Fachwissen der Hersteller. Für sie können lediglich Nutzungsrechte erworben werden, die Daten bleiben im Besitz der Anbieter. Da diese Daten hochsensibel sind, wird ein besonderer Fokus auf IT-Sicherheit und den Schutz vor unberechtigter Verwendung gelegt.

Für kleine Unternehmen mit einer begrenzten Anzahl an Laserbearbeitungsmaschinen ist die manuelle Bereitstellung, Verwaltung und Verteilung von erworbenen Technologiedaten und der zugehörigen Lizenzinformationen noch handhabbar. In großen Unternehmen, die über viele Laserbearbeitungsmaschinen und unterschiedliche Lasertechnologien von verschiedenen Herstellern verfügen, ist diese manuelle Verwaltung jedoch aufwendig und unwirtschaftlich. So ist es für Unternehmen mit einem breiten Spektrum an Laserbearbeitungsmaschinen und Technologien oft schwierig, den Überblick über erworbene Technologiedaten, die entsprechenden Laserbearbeitungsmaschinen und die zugehörigen Lizenzen zu behalten. Außerdem erhöht sich die Komplexität bei der Auswahl der geeigneten Lasermaschine für anstehende Aufträge. Eine weitere Herausforderung stellt die Auswahl der für einen spezifischen Auftrag passenden Technologiedaten-Lizenzmodelle dar. Unternehmen stehen daher vor der Herausforderung, nicht nur die Bereitstellung, Verwaltung und Verteilung der benötigten Technologiedaten effizient zu gestalten, sondern auch die sichere Nutzung und Nachverfolgung der erworbenen Technologiedaten und der zugehörigen Lizenzinformationen zu gewährleisten. Ferner bedeutet die eindeutige Zuordnung von erworbenen Technologiedaten zu den jeweiligen Maschinen und den zugehörigen Lizenzen einen erheblichen organisatorischen Aufwand.

Bisher sind keine Lösungen bekannt, die Anwender bei der effektiven und sicheren Nutzung von Technologiedatenmarktplätzen in der Lasermarkierung speziell unterstützen. Dabei fehlt es vor allem an Systemen für eine schnelle, bedarfsorientierte und effektive Bereitstellung, Verwaltung und Verteilung von Technologiedaten und der zugehörigen Lizenzinformationen. Ebenso mangelt es an Hilfestellungen für eine effiziente Auswahl passender Maschinen- und Lizenzmodelle. Lösungen zur Unterstützung des Anwenders bei der Nutzung von Technologiedatenmarktplätzen, etwa bei der Auswahl der geeigneten Laserbearbeitungsmaschinen und der passenden Lizenzmodelle für die Durchführung Kundenaufträgen mit unterschiedlichen Laserfertigungsprozessen, sind auch nicht bekannt. Voraussetzung für die Teilnahme an solchen Technologiedatenmarktplätzen ist die Gewährleistung der Datensicherheit. Die dafür erforderlichen Sicherheitsanforderungen und -maßnahmen sind noch nicht untersucht und definiert. Es besteht außerdem keine Übersicht zum Nutzungsstatus von erworbenen Technologiedaten und der zugehörigen Lizenzen.

Diese Defizite verdeutlichen die Notwendigkeit einer systematischen, methodisch fundierten Herangehensweise zur Entwicklung eines rechnergestützten Ansatzes, der den Anwendern praxisnahe Empfehlungen für die Auswahl von Maschinen und Lizenzmodellen bietet. Darüber hinaus müssen sichere Konzepte für den Schutz von Technologiedaten und der zugehörigen Lizenzinformationen sowie für die Verfolgbarkeit ihrer Nutzung entwickelt werden.

Hieraus lassen sich die folgenden Forschungsfragen ableiten, welche im Verlauf dieser Dissertation bearbeitet und durch das entwickelte Konzept beantwortet werden:

- Wie können Technologiedaten für Laserbearbeitungsmaschinen effizient und sicher über einen Technologiedatenmarktplatz vertrieben werden?

- Welche rechnergestützten Methoden eignen sich zur Bereitstellung von Handlungsempfehlungen für die Auswahl geeigneter Laserbearbeitungsmaschinen, um die Produktionsplanung zu optimieren?

- Wie kann die Lizenzmodellauswahl für spezifische Aufträge im Kontext der Lasermarkierung flexibel und effektiv gestaltet werden?

- Welche Sicherheitsmechanismen sind erforderlich, um den Schutz und die Nachverfolgung von erworbenen Technologiedaten und der zugehörigen Lizenzinformationen zu gewährleisten?

- Wie können die zwischen dem Technologiedatenmarktplatz und dem Anwender vereinbarten Nutzungsbedingungen einer Lizenz zuverlässig durchgesetzt werden?

- Wie kann die kontinuierliche Überwachung der Nutzung von erworbenen Technologiedaten effektiv implementiert werden?

- Welche Methoden eignen sich zur Darstellung von Lizenzen im Kontext eines Technologiedatenmarktplatzes?

- Wie können Technologiedaten und zugehörige Informationen systematisch verwaltet und gesichert werden?

Diese Fragestellungen werden in den nachfolgenden Kapiteln dieser Dissertation wissenschaftlich beleuchtet und mittels geeigneter Methoden und Ansätzen beantwortet.

1.2 Zielsetzung

Der wissenschaftliche Kern dieser Dissertation besteht in der Konzeption eines Ansatzes zur Unterstützung des Anwenders bei der effektiven und sicheren Nutzung von Technologiedatenmarktplätzen, exemplarisch dargestellt am Beispiel der Lasermarkierung. Dieser Ansatz bildet die Grundlage für die effiziente Bereitstellung, kontrollierte Verteilung und sichere Nutzung von über Technologiedatenmarktplätze erworbenen Technologiedaten, insbesondere in der Lasermarkierung.

Das übergeordnete Ziel der Dissertation ist die Entwicklung eines rechnergestützten Assistenzsystems, das Anwender bei der effektiven und sicheren Nutzung von Technologiedatenmarktplätzen unterstützt. Um dieses übergeordnete Ziel zu erreichen, ist es notwendig, geeignete Methoden für die Konzeption dieses sicheren Assistenzsystems anzuwenden. Dies schließt die Methoden zur Gewährleistung eines sicheren Datenaustauschs und zum Schutz der wertvollen Ressource, nämlich der erworbenen Technologiedaten, ein.

Als erstes Teilziel der Dissertation wird die Entwicklung eines Konzepts zur Maschinenauswahl angestrebt. Dieses Konzept soll dem Anwender Handlungsempfehlungen für die Auswahl geeigneter Laserbearbeitungsmaschinen für einen spezifischen Laserauftrag bereitstellen, und zwar basierend auf den Auftragsinformationen. Dabei soll auch die Verwaltung und Absicherung von Eigenschaftsdaten der existierenden Laserbearbeitungsmaschinen ermöglicht werden. Dies erfordert eine genaue und detaillierte Analyse der betreffenden Eigenschaftsdaten, die zur Beurteilung der Tauglichkeit einer Maschine für die Bearbeitung eines spezifischen Auftrags herangezogen werden. Weiterhin soll ein Ansatz definiert werden, der diese Beurteilung automatisch und systematisch durchführt. Als Ergebnis soll dem Anwender eine Liste der für den spezifischen Laserauftrag geeigneten Laserbearbeitungsmaschinen zur Verfügung gestellt werden.

Das zweite Ziel besteht darin, dem Anwender Handlungsempfehlungen bezüglich der Auswahl von Lizenzmodellen für den Bezug der benötigten Technologiedaten anzubieten. Das entwickelte Konzept dient dazu, den Auswahlprozess zu vereinfachen und effizienter zu gestalten. Zuerst sollen die Merkmale der Lizenzmodelle definiert werden, die für den Vertrieb

von Technologiedaten auf Technologiedatenmarktplätzen benötigt werden. Dies erfordert eine umfassende Analyse der bereits existierenden Lizenzmodelle für vergleichbare digitale Produkte. Die so definierten Lizenzmodelle werden dann in die Konzeption der Lizenzmodellauswahl integriert. Anschließend sollen geeignete Methoden und Mechanismen spezifiziert werden, um das Ziel – die Formulierung der besagten Handlungsempfehlungen – zu erreichen. Dafür kommen rechnergestützte Verfahren zur Verarbeitung und Bewertung der verfügbaren Informationen zum Einsatz. Diese beiden Zielsetzungen – die Entwicklung je eines Konzeptes für die Maschinenauswahl und die Auswahl von Lizenzmodellen – sollen den Anwender bei der effizienten Bereitstellung von Technologiedaten auf dem jeweiligen Technologiedatenmarktplatz für einen spezifischen Kundenauftrag unterstützen.

Die finale Entscheidung hinsichtlich der Wahl der Laserbearbeitungsmaschine und des Lizenzmodells fällt im Rahmen der Auftragsplanung des Unternehmens und ist nicht Gegenstand dieser Dissertation, da für diese Fragestellungen bereits Lösungen existieren. Der Fokus dieser Arbeit liegt vielmehr auf der Bereitstellung fundierter Handlungsempfehlungen, die die Entscheidungsfindung in diesen Bereichen unterstützen und dazu beitragen können, die Produktionskosten zu minimieren.

Das dritte Ziel dieser Dissertation besteht in der Gewährleistung der Sicherheit von Technologiedaten und Lizenzinformationen. Im Fokus stehen sowohl die sichere Übertragung und Verteilung als auch eine schnelle Verfügbarkeit der Daten für die Laserbearbeitung. Der Schutz sowohl der Technologiedaten als auch der Lizenzinformationen ist für die Konzeption von zentraler Bedeutung. Das entwickelte Konzept zielt darauf ab, dem Anwender einen sicheren Zugang zu bereits erworbenen Lizenzen zu gewähren und gleichzeitig eine überwachte Nutzung der Technologiedaten zu ermöglichen. Der vollständige Prozess von der Lizenzbeschaffung bis zur praktischen Anwendung der Technologiedaten in Laserbearbeitungsmaschinen wird dabei in Betracht gezogen, um lückenlosen Datenschutz zu garantieren. Es ist wichtig zu betonen, dass die Lizenzbedingungen unveränderlich sein müssen und nicht nachträglich modifiziert werden dürfen. Zur Umsetzung des Konzepts werden geeignete Sicherheitsmechanismen und -methoden integriert, um ein fehlerresistentes Lizenzierungskonzept für Technologiedaten zu entwickeln.

Ein weiteres zentrales Ziel dieser Dissertation besteht darin, die entwickelten Konzepte unter realitätsgetreuen Bedingungen auf ihre Sicherheit und Effizienz zu prüfen. Zu diesem Zweck ist eine prototypische Softwareimplementierung verschiedener Systemkomponenten notwendig. Das entwickelte Assistenzsystem soll dem Anwender maßgeschneiderte Funktionen und

Informationen bieten und dabei eine effektive sowie sichere Interaktion mit Technologiedatenmarktplätzen ermöglichen. Zudem ist die Entwicklung von repräsentativen Anwendungsfällen erforderlich, die typische Aufträge im Bereich der Laserfertigung simulieren und mit denen das System unter realen Bedingungen getestet wird. Diese physische Implementierung und die nachfolgende Validierung sind entscheidende Schritte, um die Einsatztauglichkeit des entwickelten Assistenzsystems zu verifizieren.

1.3 Struktur der Dissertation

Die vorliegende Dissertation zur Konzeption eines Ansatzes für die Unterstützung des Anwenders bei der effektiven und sicheren Nutzung von Technologiedatenmarktplätzen am Beispiel der Lasermarkierung ist in acht Hauptkapitel untergliedert.

In den vorhergehenden Abschnitten wurden wurde in das Thema eingeführt und es wurden die zentralen Fragestellungen sowie die daraus abgeleiteten Zielsetzungen dargelegt. In Kapitel 2 wird der Stand der Technik innerhalb der für diese Dissertation wesentlichen Themenfelder untersucht. Dies umfasst die Beschreibung wesentlicher Konzepte und die Definition der wichtigsten Fachbegriffe. Hierbei werden Methoden und Konzepte aus den Bereichen Lasertechnologie, E-Commerce, maschinelles Lernen, Modellierung, Lizenzierung und IT-Sicherheit vorgestellt. Darüber hinaus werden für die Thematik dieser Dissertation relevante Forschungsansätze diskutiert und beschrieben.

Basierend auf der formulierten Problemstellung und dem Stand der Technik wird in Kapitel 3 der wissenschaftliche Handlungsbedarf abgeleitet. Aufbauend hierauf werden konkrete Anforderungen an die Konzeption und die technische Umsetzung des entwickelten Assistenzsystems definiert. Das Ergebnis ist ein Anforderungsprofil, auf dessen Grundlage das Konzept und Methoden für dessen Validierung entwickelt werden.

In Kapitel 4, dem Kern dieser Dissertation, wird das Konzept zur Unterstützung des Anwenders bei der effektiven und sicheren Nutzung von Technologiedatenmarktplätzen in der Lasermarkierung vorgestellt. Dies umfasst im Wesentlichen die Vorstellung der drei Kernelemente: Maschinenauswahl, Lizenzmodellauswahl und Datensicherheit.

Zur Überprüfung der Tragfähigkeit des entwickelten Konzeptes wird es in Kapitel 5 in Form einer Webapplikation prototypisch implementiert.

Diese Implementierung dient als Grundlage für die Validierung und Verifikation des entwickelten Konzeptes in Kapitel 6. Hierfür wird ein repräsentativer, praxisnaher Anwendungsfall definiert. Außerdem werden die Sicherheit und die Leistungsfähigkeit des Assistenzsystems überprüft und bewertet. Anschließend erfolgt eine formale Verifikation des entwickelten Konzeptes anhand des in Kapitel 3 entwickelten Anforderungsprofils.

Zukünftige Entwicklungs- und Forschungspotenziale werden auf Grundlage der Ergebnisse dieser Dissertation als Ausblick in Kapitel 7 diskutiert.

Schließlich erfolgt eine Zusammenfassung zur vorliegenden Dissertation und der vorgestellten Ergebnisse in Kapitel 8.

2 STAND DER TECHNIK

In Zeiten der Industrie 4.0 sind die Produktionsprozesse durch eine intensive Nutzung von IKT (Informations- und Kommunikationstechnologien) intelligenter geworden. Dank dieser Technologien können alle am Fertigungsprozess beteiligten Entitäten – Maschinen, Werkstücke und Nutzer – miteinander kommunizieren, was zu einer Verbesserung der Produktqualität, zu mehr Flexibilität und zu kürzeren Produktionszeiten führen kann. Aufbauend darauf können neue datenbasierte Dienstleistungen und Geschäftsmodelle entwickelt werden. Beispielsweise können Dienstleistungen für eine vorausschauende Wartung (engl.: Predictive Maintenance) angeboten werden. Die „Industrie 4.0“ wird von der gleichnamigen Plattform wie folgt definiert:

> ***Industrie 4.0 bezeichnet die intelligente Vernetzung von Maschinen und Abläufen in der Industrie mit Hilfe von Informations- und Kommunikationstechnologie** [10].*

Dieser digitale Wandel in der Industrie umfasst auch eine wirtschaftliche Transformation hin zu die ganze Welt umspannenden digitalen Ökosystemen, wobei klassische Wertschöpfungsketten in globale, flexible und hochdynamische Wertschöpfungsnetzwerke transformiert werden. Insbesondere wird künftig zu beobachten sein, dass datenbasierte Geschäftsmodelle ein enormes Wachstum erleben werden [11]. Die Digitalisierung ermöglicht eine effektive Datenerfassung, -verarbeitung, -speicherung und -übertragung. Dafür bedarf es moderner Hardware- und Softwareanwendungen. Die steigende Vernetzung von Systemen und Menschen und die Informationsrevolution sowie die intensive Verwendung von Informations- und Kommunikationstechnologien (IKT) ermöglichen die Herausbildung einer globalen, grenzüberschreitenden Wirtschaft [12], [13].

In diesem Kapitel wird der für die Entwicklung eines sicheren Assistenzsystems zur Unterstützung des Anwenders bei der effektiven und sicheren Nutzung von Technologiedatenmarktplätzen erforderlicher Stand der Technik vorgestellt. Zuerst wird in Abschnitt 2.1 auf den E-Commerce-Trend und die zunehmende Verbreitung von E-Marktplätzen in der Industrie eingegangen. Dabei wird speziell auf die Datenmarktplätzen fokussiert. Im Abschnitt 2.2 werden die Lasertechnologie und deren Applikationen in der Industrie erläutert. Darüber hinaus wird auf die Technologiedaten eingegangen und es werden die Datenflüsse hin zu den Laserbearbeitungsmaschinen nachgezeichnet. In Unterkapitel 2.2.4 erfolgen dann noch

Erläuterungen zu den Technologiedatenmarktplätzen. Die Funktionsweise eines wissensbasierten Assistenzsystems wird in Unterkapitel 2.3 dargestellt. Als ein besonders wichtiges Thema werden in Abschnitt 2.4 auch die aktuellen und verwandten Sicherheitstechnologien aufgearbeitet. Hierbei wird vertiefend auf Kryptographie, Blockchain- und Smart-Contract-Technologien eingegangen. In Unterkapitel 2.4.3 werden die Zugriffsberechtigungen auf die Daten und die Lizenzierungen vorgestellt. Dabei werden Lizenzformen und -modelle erklärt und es wird der Einsatz von Blockchain- und Smart-Contract-Technologien in der Lizenzierung digitaler Inhalte vorgestellt. Die in dieser Dissertation verwendeten Modellierungsmethoden werden im Abschnitt 2.5 dargestellt. Das Kapitel 2 schließt mit einem Fazit in Abschnitt 2.6 ab, in dem die wissenschaftlichen Lücken, mit denen sich diese Dissertation beschäftigt, identifiziert und daraus die Potenziale und die Herausforderungen bei der Nutzung von Technologiedatenmarktplätzen in der Lasermarkierung ableitet werden.

2.1 E-Commerce und E-Marktplatz

In diesem Kapitel wird ein Überblick über den Bereich E-Commerce und über Datenmarktplätze sowie über deren Handelskonzepte gegeben. E-Commerce hat in den letzten Jahrzehnten aufgrund der Digitalisierung ein enormes Wachstum erlebt. Insbesondere der Onlinehandel mit digitalen Produkten wie zum Beispiel Musik, Filme und E-Books ist sehr stark angestiegen. Onlinemarktplätze sind Plattformen, die Händlern die Möglichkeit bieten, neue Geschäftsmodelle zu entwickeln und digitale Produkte zu erwerben oder anzubieten. Sie basieren auf aktuellen Technologien. Ein Beispiel für so ein innovatives Geschäftsmodell ist der Datenhandel mittels sicherer und stabiler Transaktionen [14], [15]. Die Entwicklung von sozialen Netzwerken wie *Google+*, *Twitter* und *Facebook* führte zur Erweiterung und zur zunehmenden Verbreitung von E-Commerce-Lösungen, die momentan eine bedeutsame Rolle in der Wirtschaft und in der Gesellschaft spielen [16].

Electronic-Commerce, meist „E-Commerce“ genannt, wurde als neuer Begriff in den Siebzigerjahren etabliert [56] und beruht auf der Verbreitung des Internets in den letzten Jahrzehnten. Es geht dabei um den Handel mit Produkten und Dienstleistungen über sogenannte „Online-Handelsplattformen“. Diese Plattformen fungieren als Schnittstellen zwischen Käufern und Verkäufern, die Personen oder Unternehmen sein können. E-Commerce wird folgendermaßen definiert:

> ***Electronic commerce is a business model in which transactions take place over electronic networks, mostly the internet. It includes the process of electronically buying and selling goods, services and information [16].***

Es gibt verschiedene E-Commerce-Konzepte zur Bindung des Endverbrauchers an die Unternehmen. Sie lassen sich im Wesentlichen anhand des Verhältnisses zwischen den Marktteilnehmern – den Transaktionsparteien – unterscheiden. Gängige Bezeichnungen für diese Konzepte lauten Business-to-Business (B2B), Business-to-Customer (B2C), Customer-to-Business (C2B) oder Customer-to-Customer (C2C) (siehe Abbildung 2.1). Bei B2B-Modellen nehmen Unternehmen oder Geschäfte, z. B. ein Lieferant und ein Unternehmen, eine Geschäftsbeziehung auf. B2C-Geschäftsmodelle konzentrieren sich auf die Bereitstellung von Dienstleistungen für Endverbraucher bzw. für private Kunden. Bei C2B-Geschäftsmodellen ist es hingegen umgekehrt: die Verbraucher initiieren eine Handelsbeziehung mit Unternehmen. Und C2C-Geschäftsmodelle ermöglichen Transaktionen zwischen Privatkunden und Endverbrauchern [15], [17].

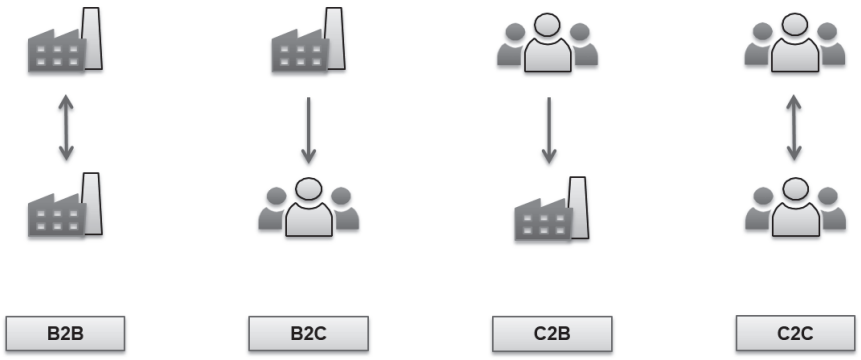

Abbildung 2-1: E-Commerce-Typen (eigene Darstellung angelehnt an [14], [15] und [17])

Davon abgesehen gibt es noch erweiterte Arten solcher Geschäftsmodelle, die aus den oben genannten abgeleitet werden, z. B. B2B2C-Geschäftsmodelle. Hierzu können Serviceplattformen, lokale Marktplätze oder Social-Commerce-Plattformen gezählt werden [14]. Auf Serviceplattformen wie z. B. *Airbnb*, *Uber* und *Soundcloud* vermieten private und gewerbliche Anbieter Dienstleistungen und Produkte. Lokale Plattformen sind Schnittstellen zwischen Anbietern und Verbrauchern in einem Einzugsgebiet. Und über soziale Netzwerke wie *Facebook*, *YouTube*, *Instagram* usw. können Produkte direkt bezogen werden [18].

Der digitale Marktplatz (engl. E-Marketplace) wird in der Literatur oft im Kontext des E- Commerce erwähnt. Er ermöglicht einen Onlineaustausch von Produkten zwischen Käufern

und Verkäufern [18]. Elektronische Marktplätze sind virtuelle Plattformen, die den Marktteilnehmern an verschiedenen Orten und zu jedem Zeitpunkt Produkte und Dienste über das Internet zur Verfügung stellen. Die wirtschaftlichen Transaktionen zwischen Anbietern und Interessenten werden auf elektronischem Wege abgewickelt [19]. Ein B2B-E-Marktplatz wird folgendermaßen definiert:

> ***B2B-E-Marktplätze sind oftmals neutrale, digitale, webbasierte Standorte, Websites oder Plattformen, auf denen Unternehmen Käufe und Verkäufe für ihre Güter und Dienstleistungen tätigen können [20].***

Online-Marktplätze für den Handel mit digitalen Inhalten haben sich in den Bereichen Unterhaltungselektronik und Multimedia in den letzten Jahren etabliert. Beispiele dafür sind *Netflix*, *iTunes* und *Amazon prime*. Dort können Verbraucher die Nutzungsrechte angebotener Produkte gegen Bezahlung erwerben. Beispielsweise können sie Musik hören oder Filme ansehen, ohne die Eigentumsrechte dafür zu besitzen. Plattformen wie *Google Play* oder der *App Store* von *Apple* bieten Apps für Smartphone-Betriebssysteme wie *Android* und *iOS* an, dank derer die Funktionalitäten eines Gerätes erweitert werden können [21]. Zudem haben sich Onlinemarktplätze zur Durchführung von komplexen Simulationen und zur Modellierung von Bauteilen als innovative Geschäftsmodelle auf dem Markt etabliert. Trotz des Erfolgspotenzials von B2B-Marktplätzen gibt es verschiedene Herausforderungen, die in der Literatur diskutiert werden. Beispielsweise werden in [22] Fragen zur Datensicherheit und zur Integration von Marktplätzen in die internen Informationssysteme von Unternehmen aufgeworfen.

2.1.1 Datenmarktplätze

Im vorhergehenden Unterkapitel 2.1 wurden E-Commerce und E-Marktplätze allgemein beschrieben. Ferner wurden die gängigen Geschäftsmodelle von Marktplätzen dargestellt. Es folgen nun mehrere Beispiele für Marktplätze mit unterschiedlichen Ansätzen und Funktionalitäten, insbesondere stehen nun Marktplätze für digitale Daten im Fokus.

Datenmarktplätze sind seit einigen Jahren zunehmend verbreitet. Auf solchen Marktplätzen können Daten gesammelt, verarbeitet, angereichert, gekauft und verkauft werden. Durch die elektronische Bereitstellung von datenbasierten Produkten und Diensten aus vielen Quellen über das Internet können die Personal-, Transport- und Materialkosten reduziert werden, es können mehr Kunden erreicht und neue Märkte erschlossen werden [52]. Es folgen einige Definitionen des Begriffs „Datenmarktplatz“:

> ***Datenmarktplätze sind elektronische Marktplätze, auf denen verschiedene Daten als Informationsgüter gehandelt werden [23].***

Die Autoren in [24] betonen in ihrer Definition die Bedeutung von Sicherheit und Vertrauen bei der Entwicklung von Datenmarktplätzen:

> ***Ein Datenmarktplatz ist eine Plattform für Datenhandel zwischen mehreren Käufern und Verkäufern, die sichere, ehrliche und vertrauenswürdige Transaktionen gewährleistet [24].***

Datenmarktplätze können unter anderem anhand des Geschäftsmodells, der angebotenen Daten, der Funktionalität und der Marktmechanismen voneinander unterscheiden werden [25]. Die auf Datenmarktplätzen angebotenen Güter können zum einen die Daten selbst und zum anderen datenbezogene Dienste sein [23]. Bei den Daten kann es sich zum Beispiel um STL-Dateien für additive Fertigungsmaschinen handeln. Die Struktur und die Funktionsweise solcher Datenmarktplätze entsprechen den bekannten Konzepten von Markplätzen für physische Produkte [18].

Daten stellen eine wesentliche Quelle für Wertschöpfung dar, denn daraus können neue Erkenntnisse gewonnen werden, die schließlich zur Optimierung der Geschäftsprozesse im Unternehmen dienen. Auf Basis der Analyse und Auswertung von gesammelten Daten können auf Marktplätzen datenbezogene Dienste oder neue Produkte angeboten werden. Auf dem Datenmarktplatz werden beispielsweise Nutzungsdaten von verschiedenen Kunden angeboten und gesammelt [26], [27] und [28]. Somit sind Daten die Treiber von vielen innovativen Geschäftsmodellen, die in der Zukunft zunehmend von Bedeutung sein werden [29]. Ein neues Geschäftsmodell könnte auf Basis der gesammelten Daten entstehen, indem ein Dienstleister Algorithmen auf dem Datenmarktplatz hochlädt, um Daten zu analysieren und auf dieser Grundlage neue Dienste anzubieten [25].

Datenmarktplätze können entweder als zentrale oder dezentrale Plattformen entwickelt werden. Zentrale Marktplätze verfügen über eine Datenbank zur Sammlung der gehandelten Daten. Dabei werden die Teilnehmer und Transaktionen von einer zentralen Stelle verwaltet. Bei dezentralen Marktplätzen speichern die Nutzer ihre Daten auf ihrem eigenen Gerät und haben deshalb die volle Kontrolle über sie. Außerdem gibt es keine zentrale Stelle, die den Marktteilnehmer kontrolliert und verwaltet [24].

Beispiele für Datenmarktplätze

Ein Beispiel für einen Datenmarktplatz im industriellen Fertigungsbereich ist ein Production-as-a-service-Marktplatz namens *UberManufacturing*. Die Kunden können hier qualifizierte Anfragen zu gewünschten Produkten erstellen. Zudem bieten die Betreiber an, neue Produkte für die Kunden zu fertigen. Der Kunde nimmt das passende Angebot an, und dann werden die Produkte vom jeweiligen Betreiber produziert und geliefert [30].

Im Rahmen des Projektes *AutoMat (Automotive Big Data Marketplace for Innovative Cross-sectorial Vehicle Data Services)*, das von *European Union's Horizon 2020 (H2020)* gefördert wird, wurde eine zentrale Plattform für das Sammeln von und den Handel mit Fahrzeugsensordaten konzipiert [31]. Der Marktplatz übernimmt eine Vermittlerrolle zwischen dem Fahrzeugbesitzer als Datenanbieter und dem Dienstleister. Nutzungsdaten aus verschiedenen Fahrzeugsensoren werden gesammelt, vereinheitlicht, aufbereitet und in der Cloud in einem personalisierten Speicher aufbewahrt. Hierbei wird ein herstellerübergreifendes Datenmodell, das *Common Vehicle Information Model* (CVIM) angeboten, das die Aggregation von markenunabhängig erfassten Datensätzen ermöglicht. Fahrzeugbesitzer können die Sammlung der Fahrzeugsensordaten genehmigen und behalten gleichzeitig die volle Kontrolle über ihre Daten. Diese werden aufbereitet, indiziert und zur Verfügung gestellt. Zulieferer können dann Datenanfragen auf dem Marktplatz erstellen und basierend auf den erfassten Datensätzen neue Erkenntnisse gewinnen, um innovative Dienstleistungen und Applikationen zu entwickeln. Details zu diesem Konzept finden sich in [32].

In [33] wird ein weiteres Beispiel für einen offenen und dezentralen Marktplatz angeführt, auf dem landwirtschaftliche Dienstleistungen angeboten werden. Hier treffen Teilnehmer, Anbieter und Kunden mit kompatiblen Interessen aufeinander, und es können Dienste und Daten im Bereich Landwirtschaft angeboten und gezielt ausgetauscht werden. Es handelt sich um eine offene Plattform für einen freien Handel ohne zentrale Regulierung. Die Anbieter behalten ihre Daten auf ihrem eigenen Speicher und nur der öffentliche Teil des Angebots sowie die Teilnehmerdaten werden zentral gesichert. Bei einer erfolgreichen Suchanfrage und nachdem das Einverständnis dazu erteilt wurde, wird die Identität des Anfragenden dem Anbieter preisgegeben. Der Handel kann bei Interesse über eine P2P-Verbindung vonstattengehen, sodass die Sicherheit der gehandelten Daten garantiert ist.

Ein Konzept zu einem semi-dezentralisierten Datenmarktplatz basierend auf Smart-Contract- und Blockchain-Technologien wurde in [34] untersucht. Die Herausforderung beim Handel mit

digitalen Daten besteht zum einen darin, dass die Umtauschrichtlinien schwer zu bestimmen sind, da die Daten nach dem Öffnen und Lesen ihren Wert verlieren. Zum anderen muss die Sicherheit der Datensets sowohl bei der Lagerung als auch beim Austausch gewährleistet sein, sodass die Anbieter die volle Kontrolle über ihre Daten behalten. Die Daten werden daher vom Anbieter zum Käufer nach Abschluss des Kaufprozesses über sichere Internetkanäle übermittelt. Darüber hinaus ist die korrekte Beschreibung der angebotenen Datensets von großer Bedeutung, um bei den Nutzern keine falschen Erwartungen zu wecken.

In [34] werden funktionale und nicht funktionale Anforderungen für solche Datenmarktplätze adressiert, was wichtig ist für die Entwicklung eines vertrauenswürdigen und sicheren Datenmarktplatzes. Dabei wird auch der Einsatz von Smart-Contract- und Blockchain-Technologien in den Blick genommen.

Wibson ist ein dezentraler Datenmarktplatz, auf dem Benutzer durch das anonyme Anbieten ihrer persönlichen Daten finanzielle Vorteile erzielen können, ohne ihre Privatsphäre zu beeinträchtigen. Der Marktplatz basiert auf Blockchain-Technologie, die den Einsatz von Smart-Contracts in einer sicheren und vertrauenswürdigen Umgebung ermöglicht und unterstützt. Dadurch kann die Qualität der Daten sichergestellt und deren Herkunft verifiziert werden [24].

Im Bereich Internet of Things (IoT) wurden in den letzten Jahren bereits zahlreiche Konzepte für Datenmarktplätze für Smart Communities und Smart Cities veröffentlicht. In [35] wird beispielsweise das I3-Konzept vorgestellt, bei dem Datenbesitzer über eine einzige Applikation festlegen können, zu welchen Bedingungen und Preisen sie ihre Gerätedaten in Echtzeit übertragen bzw. verkaufen möchten. Die zuvor präsentierten Datenmarktplätze unterstreichen einerseits die wesentliche Bedeutung der IT-Sicherheit und andererseits den Einsatz moderner Smart-Contract- und Blockchain-Technologien in ihrer Entwicklung und Umsetzung.

2.2 Lasertechnologie

Der Begriff „Laser“ ist ein Akronym von „Light Amplification by Stimulated Emission of Radiation“ (dt. „Lichtverstärkung durch stimulierte Emission“) [36]. Der Emissionsprozess wird von außen künstlich angeregt bzw. stimuliert, indem ein Photon in ein angeregtes Atom hineingepumpt und dadurch ein zweites, gleichartiges Photon herauslöst wird, welches als „Duplikation“ bezeichnet wird. Grundsätzlich ist ein Laser ein Gerät, das ein verstärktes Licht aussendet [37]. Ein Laserstrahlwerkzeug kann für die Herstellung verschiedener Produkte

eingesetzt werden, wobei im Rahmen unterschiedlicher Fertigungsverfahren Werkstoffe mittels präzise eingebrachter Wärmeenergie bearbeitet werden [38].

Die Laserstrahlung ist ein Träger elektromagnetischer Energie, die ins Werkstück eingebracht wird. In der Wechselwirkungszone wird diese Energie absorbiert, was abhängig von den Strahleneigenschaften, der Einwirkzeit und den Materialeigenschaften zur gezielten Erwärmung und Verformung des Werkstücks führt: es heizt sich auf, schmilzt und/oder verdampft oder es bildet sich Plasma [38].

2.2.1 Laserapplikationen in der Fertigung

Für die Bearbeitung von metallischen und nichtmetallischen Materialien sowie zum Zusammenfügen von Kunststoffen wird in der industriellen Fertigung oftmals auf Lasertechnologie zurückgegriffen. Ein Laser hat relativ hohe Investitionskosten, zeichnet sich allerdings auch durch eine hohe Prozessgeschwindigkeit und Bearbeitungsqualität aus [39]. Der Laser kommt in verschiedenen Fertigungsschritten zum Einsatz. Zum einen kann ein Laser als Hauptwerkzeug eingesetzt werden anstelle von konventionellen Fertigungstools [40]. Im Vergleich zu konventionellen Verfahren hat der Laser den Vorteil, dass es keinen Werkzeugverschleiß gibt. Darüber hinaus kann ein weites Spektrum von Metallen, Gläsern, Keramiken und Kunststoffen mit dem Laser bearbeitet werden [36]. Zum anderen kann der Laser in die vorhandenen Werkzeugmaschinen integriert werden, womit zusätzliche Fertigungsschritte vollzogen werden können, wie zum Beispiel das Härten von Oberflächen im Zuge eines Nachbearbeitungsprozesses. Ferner kann der Laser zur Erhöhung der Prozesseffizienz eingesetzt werden – zum Beispiel können mit dem Laser schwer zu bearbeitende Materialien geschnitten werden. Das Integrieren von Lasertechnologie in bestehende Maschinen hilft, Kosten zu reduzieren, denn damit können mehrere Fertigungsprozesse bzw. Bearbeitungsschritte unmittelbar nacheinander, ohne Pausen durchgeführt werden [40]. Davon abgesehen gibt es weitere wirtschaftliche Vorteile, wie beispielsweise die Reduzierung von Durchlauf- und Rüstzeiten [36].

Lasertechnologie wird schon heute in der industriellen Produktion vielfältig verwendet, etwa zum Schneiden, für die Oberflächenbehandlung, für das Rapid Prototyping, das Abtragen und Bohren und zum Zusammenfügen von Materialien. Die Bestrahlungszeit und die Leistungsdichte müssen bei der Laserbearbeitung für jedes Bearbeitungsverfahren und -material individuell festgelegt werden [36]. Im Folgenden werden die Laser-Fertigungsverfahren, die auf der untenstehenden Abbildung 2-2 dargestellt sind, näher beschrieben. Diese Darstellung

umfasst nur beispielhaft einige Verfahren und erhebt keinen Anspruch auf Vollständigkeit, da den laufend sich verändernden Marktbedürfnissen entsprechend stetig neue Verfahren entwickelt werden.

Das heutzutage in der industriellen Fertigung am weitesten verbreitete laserbasierte Verfahren ist das Laserschneiden. Das Laserschneiden ist aus technischer Sicht ein relativ einfaches Verfahren, das sehr flexibel ist und schnell vonstattengeht [36]. Es gibt verschiedene Schneideverfahren, die je nach Zustand der Wirkungszone im Werkstück (flüssig, oxidiert oder gedampft) und nach Austreibung des Fugenwerkstoffs (diesbezüglich sind zum Beispiel das Schmelz- und Brennschneiden zu nennen) gewählt werden [38]. Der Laser wird in der Oberflächenbehandlung zum Verändern der Stoffeigenschaften eingesetzt. So werden die optischen und mechanischen Eigenschaften von Bauteilen verbessert [36]. Zum einen werden dadurch thermische oder mechanische Vorgänge ausgelöst, wie beim Härten. Zum anderen werden während der Lasereinwirkung zusätzliche Stoffe auf die Oberfläche aufgebracht, etwa beim Beschichten [38].

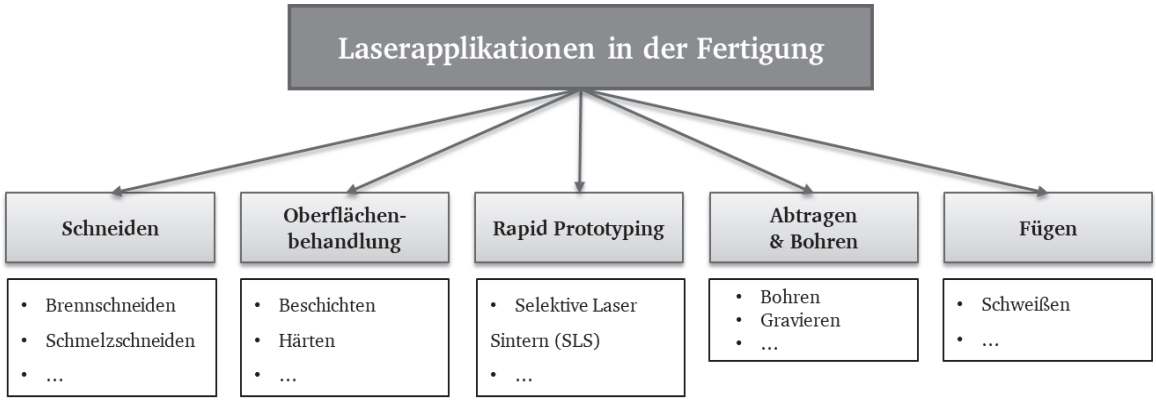

Abbildung 2-2: Laserapplikationen in der Fertigung (eigene Darstellung angelehnt an [36], [38], [41] und [42])

Rapid Prototyping (RP) ist eine Technologie zur Generierung von physischen Objekten aus grafischen digitalen Daten, zum Beispiel mittels einer CAD-Datei. Hierbei kommen keinerlei konventionelle Werkzeuge zum Einsatz. Rapid Prototyping wird sehr oft in der Automobilindustrie, in der Luft- und Raumfahrt oder in der Biomedizin sowohl in der Konstruktions- als auch in der Fertigungsphase verwendet [41]. Das Selective Laser Sintering (SLS) ist ein pulverbasiertes Verfahren. Viele Arten von Materialien in Pulverform wie zum Beispiel Thermoplast- und Verbundwerkstoffe, Keramiken und Metalle lassen sich mit diesem Verfahren zu neuen Bauelementen verarbeiten [42].

Beim Bohren und Abtragen wird Material vom Werkstück entfernt, wobei die Laserenergie vom Werkstück absorbiert wird und zur Erwärmung, zum Schmelzen und teilweise zur Verdampfung von Material führt. Die Schmelze wird durch Fliehkraft, elektromagnetische Kraft oder durch eine externe Gasströmung ausgetrieben. Zum Beispiel wird das Einzelpulsbohren für die Herstellung der Siebfilter für Dieseleinspritzeinheiten eingesetzt, wobei das Werkstück mit hoher Geschwindigkeit senkrecht zur Laserstrahlung geführt wird. Das Laserabtragen ist ein flexibles Verfahren, mit dem verschiedene Materialien im Millimeter- bis Mikrometerbereich verarbeitet werden können. Dieser Prozess wird für das Beschriften, Gravieren, Ausbleichen und Reinigen eingesetzt. Das Verfahren findet eine sehr breite Anwendung in der Elektro- und Elektronikindustrie, wo fein- und mikrotechnische Prozesse vonstattengehen [38]. Lasermarkiersysteme sind in der Industrie ebenfalls sehr verbreitet. Sie ermöglichen verschiedene Arten von Beschriftungen diverser Materialien. Zum Beispiel werden so Produkte oder Bauteile gekennzeichnet [36].

Der Laserstrahl wird auch zum Zusammenfügen von Materialien verwendet. Beim Schweißen werden mehrere Teile durch Laserenergie erhitzt, bis sie schmelzen und flüssig werden, um sie dann zu verbinden. Das Laserschweißen zeichnet sich durch schmale Schweißnähte, eine hohe Geschwindigkeit und eine geringe thermische Belastung aus [38]. Laser werden auch für das Schweißen von Karosserieteilen in der Automobilindustrie eingesetzt. Das bringt wirtschaftliche Vorteile mit sich, insbesondere können Material- und Herstellungskosten gespart werden [43]. Außerdem ermöglicht das Laserschweißen in der Automobilindustrie ein präzises Vorgehen und es können schwer zugängliche Stellen (zum Beispiel in Rohren) bearbeitet werden [36].

Abhängig vom Anwendungsbereich, vom Material und von den Verfahren werden verschiedene Lasertypen mit unterschiedlichen Eigenschaften verwendet. Gaslaser, Festkörperlaser, Diodenlaser, Faserlaser, Excimer-Laser und Farbstofflaser sind im industriellen Umfeld schon weit verbreitet [38]. Zum Beispiel findet der CO2-Laser mit Leistungen von 0,5 bis 20 kW zum Schneiden und Schweißen von Metallen regelmäßig Anwendung [40]. Es sei an dieser Stelle auf die umfangreiche Literatur über Lasertheorie, Betriebsarten, Typen, Moden und Applikationen hingewiesen, wie zum Beispiel [36], [37], [39] und [41]–[50]. Im Folgenden liegt der Fokus auf drei Laserverfahren: Laserschneiden, Laserschweißen und Lasermarkieren.

2.2.2 Technologiedaten

Über Technologiedaten, auch „technologische Daten“ genannt, ist in der Fachliteratur in Zusammenhang mit in der Produktion eingesetzten Werkzeugmaschinen die Rede. Eine

einheitliche, allgemein gültige Definition von „Technologiedaten“ existiert nicht, vielmehr wird der Begriff auf unterschiedliche Weise verwendet und beschrieben. Nach [52] umfassen Technologiedaten Angaben zu Werkstückeigenschaften wie Material, Oberflächengüte und maximal zulässige Toleranzen. Die Autoren in [53] fassen den Begriff weiter, indem sie darunter „Werkzeuge, Werkstoffe, Schnittwerte und Arbeitsabläufe“ subsumieren. Die Autoren in [54] stellen Technologiedaten als einen Sammelbegriff für verschiedene Formen von Prozess-, Produkt- und Maschinendaten dar, die entweder zur Einrichtung der Produktion gebraucht werden oder die bei der Produktion anfallen und aufgezeichnet werden.

Bei numerisch gesteuerten Werkzeugmaschinen findet eine Integration von Technologiedaten in den NC- Programmen (das sind Steuerprogramme) statt. Steuerinformationen, die zur Bearbeitung eines Werkstückes an einer NC- Anlage erforderlich sind, werden also in einem NC- Programm zusammengefasst. Der gesamte Arbeitsablauf der Maschinen wird in eine Abfolge von Bearbeitungsabschnitten gegliedert, die in Form von Befehlssätzen aneinandergereiht werden [55].

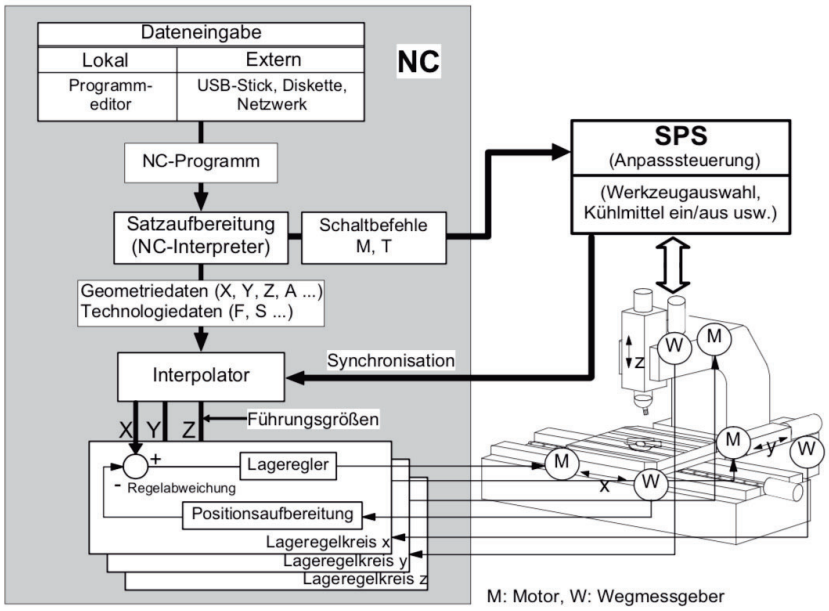

Abbildung 2-3: NC-Programm abgeglichen mit der Werkzeugmaschine (Quelle [56])

Die Autoren von [56] beschreiben den Ablauf eines NC- Programms en Detail (siehe Abbildung 2-3), wobei Technologiedaten hier konkret als Informationen dargestellt werden, die für Funktionen zur Auswahl des Werkzeugs, der Spindeldrehzahl oder der Schnittgeschwindigkeit benötigt werden.

Technologiedaten enthalten Programmanweisungen wie zum Beispiel den Befehl zum Anfahren im Eilgang. Sie werden zur Verwaltung des NC- Programms verwendet. Geometriedaten bilden die Fahrbewegungen des Werkzeugs zur Erzeugung des Bauteils ab. Sie enthalten Informationen zur Bauteilgeometrie, die der Fertigungszeichnung entnommen werden. Die Werkzeugbahn wird durch Technologie- und Geometriedaten bestimmt. Die Technologiedaten umfassen abhängig von der Fertigungsmaschine und vom Werkstück auch die Steuerungsdaten, die für die korrekte Durchführung der einzelnen Fertigungsschritte benötigt werden. Sie beinhalten darüber hinaus Informationen zur Prozesssteuerung und Werkzeugeinstellung, z. B. zur Geschwindigkeit und Beschleunigung und zum Werkzeugabstand.

In der Laserbearbeitung sind Technologiedaten ein Teil des Steuerungsprogramms, sie umfassen Informationen zur Einstellung des Laserstrahls, des Werkzeug und der Achsen. Technologiedaten werden im Rahmen des IUNO-Forschungsprojektes als eine Sammlung von Daten zur Durchführung und Steuerung des eigentlichen technologischen Bearbeitungsprozesses definiert, die benötigt werden, um mit Hilfe von Werkzeugmaschinen Werkstücke zu fertigen [57].

Technologiedaten enthalten dementsprechend Parameter, mit denen eine Maschine für eine bestimmte Art der Bearbeitung eines bestimmten Materials konfiguriert werden kann. Bei den Laserschneidemaschinen sind dies beispielsweise der Fokuspunkt oder die Schneidegeschwindigkeit [58]. In dieser Dissertation werden Technologiedaten für NC-Laserbearbeitungsmaschinen wie folgt definiert:

> ***Technologiedaten sind Datensätze zur Steuerung eines Laserbearbeitungsprozesses. Sie werden in das Steuerungsprogramm integriert und beinhalten Informationen zur Laserbearbeitungsmaschine, zum verwendeten Werkzeug, zum Werkstück und zur Prozessgestaltung bzw. zu den technologischen Arbeitsabläufen.***

Auf Abbildung 2-4 werden die Technologiedaten für die Laserbearbeitung in vier Kategorien eingeteilt.

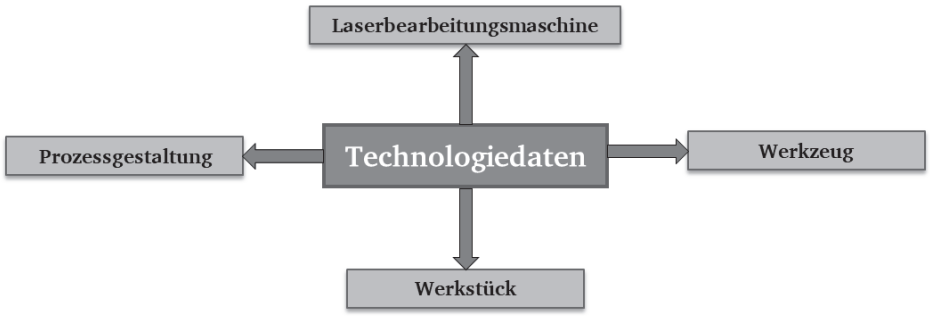

Abbildung 2-4: Technologiedaten in der Laserbearbeitung (eigene Darstellung angelehnt an [38])

Die Materialien und deren physikalische Eigenschaften wie beispielsweise die Wärmeleitfähigkeit und das Reflexionsverhalten spielen bei der Laserbearbeitung eine wichtige Rolle. In den Technologiedaten werden ebenfalls Parameter der Laserbearbeitungsmaschinen festgelegt. Dazu gehören Informationen über den Maschinentyp, die Technologie und die Optik. Bei der Laserbearbeitung ist die Auswahl des Werkzeugs entscheidend, hierbei werden zum Beispiel die Linse und die Brennweite festgelegt. Unter dem Begriff „Prozessgestaltung“ werden die Betriebsart, die Bearbeitungsstrategie, die Konturgrößen, die Qualität, die Geschwindigkeit und die Fokuslage gefasst [38]. Für den praktischen Einsatz der Lasertechnologie sind Kenntnisse und Wissen über die richtige Festlegung dieser Parameter notwendig, um befriedigende Anwendungsergebnisse zu erzielen [36].

Für die Bearbeitung verschiedener Werkstoffe auf Laserbearbeitungsmaschinen sind unterschiedliche Werkzeugeinstellungen sowie Prozessparameter notwendig. Durch die Änderung der Technologiedaten können die Prozess- und die Produktqualität sowie die Bearbeitungsgeschwindigkeit beeinflusst werden. Um bestehenden Qualitätsanforderungen bei der Laserbearbeitung gerecht zu werden, müssen die Laserparameter korrekt eingestellt werden. Die richtige Einstellung dieser Parameter ermöglicht ein optimales Prozessergebnis, wobei langjährige Erfahrung, ausführliche technologische Kenntnisse und ein tiefgreifendes Wissen des Maschinenbetreibers bzw. -bedieners von großem Vorteil sind.

2.2.3 Datenfluss zur Maschinensteuerung

Eine Laserbearbeitungsmaschine besteht grundsätzlich aus einer Benutzungsschnittstelle, einem Steuerungssystem und einem Datenbanksystem zur Speicherung der Daten. Für die Durchführung eines Laserbearbeitungsprozesses wird ein numerisch gesteuertes Programm (das NC-Programm) benötigt. Dieses Programm verarbeitet die Geometriedaten des Werkstücks

sowie die Technologiedaten. Auf der Abbildung 2-5 sind die Kommunikations- und Datenflüsse zur Steuerung der Laserbearbeitungsmaschine dargestellt. In das Steuerungsprogramm werden Parameterwerte zur Steuerung des Laserstrahls, des Achssystems und des Werkzeugs bzw. des Laserbearbeitungskopfes eingespeist.

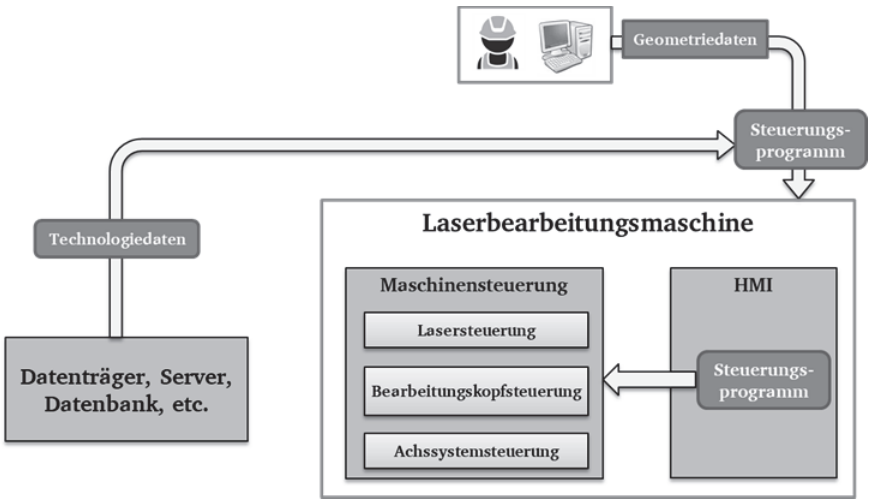

Abbildung 2-5: Datenfluss zur Maschinensteuerung (eigene Darstellung)

Die Technologiedaten sind für die Einstellung des Lasers erforderlich. Sie sind für ein bestimmtes Werkstück, einen Maschinentyp, eine Technologie und einen Bearbeitungskopf gültig. Die Technologiedaten können in der Datenbank der Laserbearbeitungsmaschine, auf einem Server oder auch auf einem externen Datenträger gespeichert werden. Sie werden über die Bedienoberfläche von autorisierten Nutzern abgerufen und können beim Bedarf angepasst werden. Die Geometriedaten beziehen sich auf das herzustellende Bauteil, sie werden zur Steuerung der Werkzeugbahn der Laserbearbeitungsmaschine benötigt. Diese Datei kann zum Beispiel auf einem CAD/CAM-Arbeitsplatz erstellt werden und wird im Anschluss zusammen mit den Technologiedaten an die Maschinensteuerung geschickt, um das Fertigungsverfahren starten zu können.

2.2.4 Technologiedatenmarktplätze

Das Konzept Technologiedatenmarktplatz wurde im Rahmen des Nationalen Referenzprojekts zur IT-Sicherheit in der Industrie 4.0 mit dem Titel *IUNO*, das vom *Bundesministerium für Bildung und Forschung* (BMBF) gefördert wurde, untersucht und bearbeitet [59]. Das Thema ist dem Forschungsgebiet Entwicklung von neuen und innovativen Geschäftsmodellen in der

vernetzten und digitalen Industriewelt zuzuordnen. Im Rahmen dieses Projekts wurde schwerpunktmäßig auf IT-Sicherheitsmaßnahmen eingegangen.

Ein Technologiedatenmarktplatz ist ein Marktplatz, über den Technologiedaten als digitale Handelsware angeboten werden. Technologiedatenmarktplätze leisten einen wichtigen Beitrag zur Erhöhung der Flexibilität und der Kosteneffizienz der Anwender in der Laserbearbeitung [60]. Dank innovativer Geschäftsmodelle dieser Art können Systeme und Akteure über die Unternehmensgrenze hinweg vernetzt werden, womit neue Werte generiert werden. Die Unternehmen kommunizieren auf diesem Weg miteinander und tauschen Daten aus, was jedoch nicht nur Vorteile, sondern auch Gefahren hinsichtlich der Datensicherheit mit sich bringt. Im besagten Forschungsprojekt wurden Sicherheitsmaßnahmen für einen Technologiedatenmarktplatz untersucht, entwickelt und prototypisch für ein Flüssigkeitsmischgerät implementiert. Hierbei stand der Schutz von Technologiedaten aus Sicht des Maschinenherstellers bzw. des Anbieters von Technologiedaten und des Marktplatzbetreibers im Fokus.

Zunächst wurden die Vermögenswerte sowie deren Bedrohung – also die Risiken – methodisch und systematisch identifiziert, analysiert und bewertet. Die identifizierten Vermögenswerte sind in diesem Zusammenhang die Technologiedaten, da sie geistiges Eigentum des Herstellers sind, in das Know-how eingeflossen ist. Der Schutz der Technologiedaten ist eine wesentliche Voraussetzung für den Erfolg des Geschäftsmodells. Verschiedene Lösungsansätze und Maßnahmen zum Schutz von Technologiedaten wurden in den Blick genommen. In der Folge wurde ein Technologiedatenmarktplatz für Getränkerezepte entwickelt und exemplarisch implementiert. Dieser kann über den Link http://iuno.axoom.cloud/ besucht werden [61].

In [62] werden alle potenziellen Teilnehmer eines solchen Technologiedatenmarktplatzes charakterisiert. Es kann sich dabei um Personen oder Systeme, beispielsweise um Maschinenhersteller, Zulieferer, Forscher, Kunden oder auch Maschinen handeln. Ferner werden ihre Interessen, ihre Bedürfnisse und Ziele in diesem Kontext diskutiert und näher beschrieben. Dazu werden Anwendungsfälle, das Systemverhalten und die Systemgrenzen für die Entwicklung eines sicheren Konzeptes für den Onlinehandel mit Technologiedaten im globalen Fertigungsmarkt erörtert. In einer Folgeveröffentlichung [63] werden die Anforderungen, welche verschiedene Szenarien an die unterschiedlichen Lizenzmodelle stellen, sowie die Arbeitsabläufe beim Anbieten und Erwerben von Technologiedaten über den Technologiedatenmarktplatz dargestellt und beschrieben. Auf dem Marktplatz können von Laserbearbeitungsmaschinenherstellern Technologiedaten mittels verschiedener Kaufmodelle

angeboten werden. Das Angebot beinhaltet Informationen zu benötigten Lasereinstellungen – genauer gesagt Technologiedaten für die Durchführung eines Laserbearbeitungsprozesses für ein bestimmtes Material, eine Prozessspezifikation und einen Maschinentyp. Ferner werden Informationen über die Qualität und die Wirtschaftlichkeit bzw. die Kosten des Prozesses angeboten. Zusammengefasst soll das vorgestellte Konzept es ermöglichen, Technologiedaten von verschiedenen Maschinenbetreibern kostengünstig und mit einer bedarfsgerechten Lizenz über den Technologiedatenmarktplatz schnell, flexibel und sicher zu erwerben. In Abbildung 2- 6 wird das Konzept für den Onlinedatenhandel noch einmal schematisch dargestellt. Das Sicherheitsrahmenwerk für den Austausch der Daten zwischen dem Technologiedatenmarktplatz und der Maschine wurde gleichfalls untersucht und definiert. Die wichtigsten Schutzziele bezogen auf die Vertrauensgrenzen der Teilnehmer (Technologiedatenhersteller, Technologiedatenmarktplatz und Maschinenbetreiber) sind ebenfalls dargestellt. Grundsätzlich sollen Integrität, Vertraulichkeit, Authentizität und Verbindlichkeit sichergestellt werden. Ferner sollen die Daten bei Bedarf unmittelbar zur Verfügung stehen.

Darüber hinaus werden in obiger Abbildung die Kommunikationswege zum Datenaustausch und die erforderlichen Schutzmaßnahmen veranschaulicht. Die dargestellten Sicherheitsmaßnahmen und die Ansätze zur Gewährleistung eines sicheren Übertragungspfads von Technologiedaten über die gesamte Wertschöpfungskette, beginnend beim Technologiedatenmarktplatz bis zur Laserbearbeitungsmaschine beim Maschinenbetreiber, werden in [64] definiert und beschrieben.

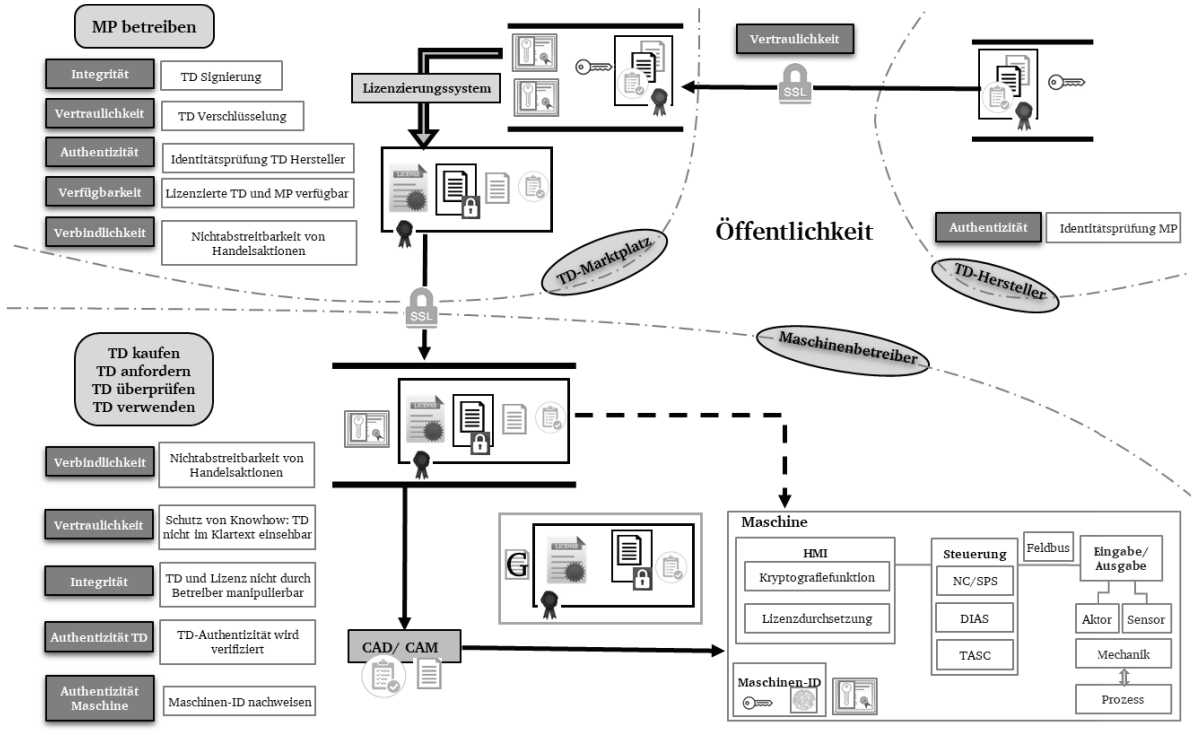

Abbildung 2-6: Sicherheitsrahmenwerk für den TD-Austausch zwischen Technologiedatenmarktplatz und Maschine (Quelle [64])

Technologiedaten dürfen nicht ohne Berechtigung genutzt werden. Deshalb werden sie von den Technologiedatenmarktplätzen in verschlüsselter Form an den Anwender geschickt. Die Internetverbindung zwischen den Technologiedatenmarktplätzen und dem Anwender muss sicher sein. Die verschlüsselten Technologiedaten werden bei Bedarf mit Hilfe der dazugehörigen Lizenz beim Anwender entschlüsselt und dann an die Laserbearbeitungsmaschinen geschickt. Um sicherzustellen, dass die Kaufbedingungen erfüllt werden, wurde in [65] eine bestehende Industrielösung von *Wibu Systems* namens „CmStick“ in der Implementierung verwendet. Auf diesem Stick wurde eine entsprechende Lizenz gespeichert, die den Zugriff auf Technologiedaten bzw. auf Getränkerezepte nur dann ermöglichte, wenn der Stick an die Maschine angeschlossen war. Hierbei handelt es sich um eine hardwarebasierte Lizenzierung.

2.3 Wissensbasierte Assistenzsysteme

Ein wissensbasiertes Assistenzsystem ist eine Art von künstlicher Intelligenz (KI), die Wissen und Regeln nutzt, um komplexe Probleme in einem bestimmten Bereich zu bewältigen [66]. Mit einem solchen System wird die Entscheidungsfindung eines menschlichen Experten in

einem bestimmten Anwendungsbereich, z. B. Medizin, Finanzen oder Technik, nachgebildet [67].

Wissensbasierte Assistenzsysteme bestehen in der Regel aus drei Hauptkomponenten: einer Wissensbasis, einer Inferenzmaschine und einer Benutzerschnittstelle (siehe Abbildung 2-7). Die Wissensbasis (engl. Knowledge Base) enthält Informationen über den betreffenden Bereich, einschließlich Fakten, Regeln und Heuristiken, und die Inferenzmechanismus (engl. Inference Engine) nutzt dieses Wissen, um Entscheidungen zu treffen und Probleme zu lösen. Die Benutzerschnittstelle (engl. User Interface) ermöglicht es Anwendern, mit dem System zu interagieren und auf der Grundlage ihrer Eingaben Empfehlungen oder Lösungen zu erhalten.

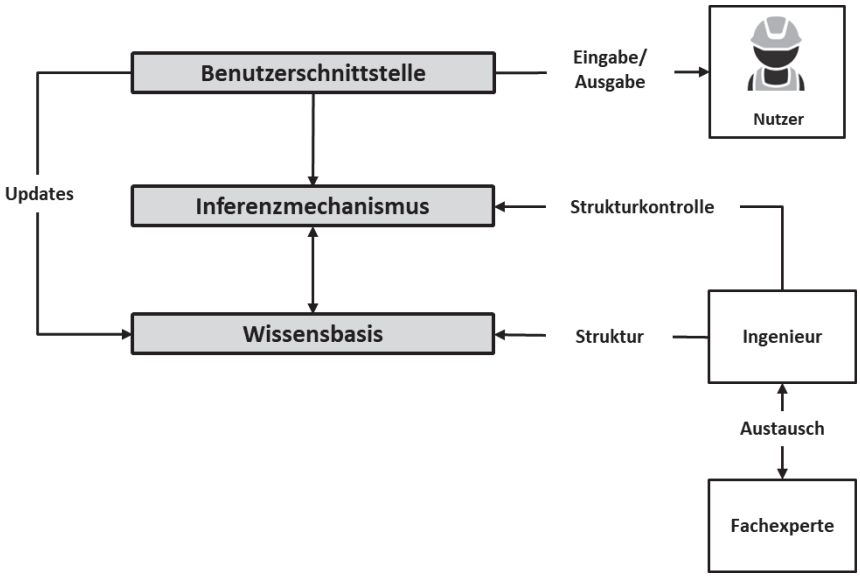

Abbildung 2-7: Wissensbasiertes Assistenzsystem Komponente (eigene Darstellung angelehnt an [59] und [67])

In wissensbasiertem Assistenzsystemen gibt es in der Regel drei Hauptrollen: den Fachexperten, den Ingenieur und den Nutzer.

1. Der Fachexperte: Der Fachexperte repräsentiert eine Person, die über tiefgreifendes Wissen und umfassende Erfahrung in einem spezifischen Fachgebiet verfügt. Diese individuelle Quelle trägt maßgeblich zur Ausgestaltung der Wissensbasis des Systems bei, indem sie komplexe Regeln, empirische Fakten sowie heuristische Ansätze liefert. Diese Elemente dienen dem System als Grundlage zur Entscheidungsfindung. Der Fachexperte arbeitet in enger Abstimmung mit dem Ingenieur, um die ständige Aktualität der Wissensbasis im Einklang mit den aktuellen Entwicklungen der jeweiligen

Fachdisziplin sicherzustellen und somit die Erzielung angemessener Ergebnisse zu gewährleisten.

2. Der Ingenieur: Der Aufgabenbereich des Ingenieurs erstreckt sich über den Entwurf, die Entwicklung und die kontinuierliche Pflege des Assistenzsystems. In enger Zusammenarbeit mit dem Fachexperten schafft der Ingenieur die Struktur der Wissensbasis, entwickelt die Inferenzmaschine und gestaltet die Benutzeroberfläche des Systems. Sorgfältige Tests und Validierungen unterliegen seiner Verantwortung, um sicherzustellen, dass das System präzise und zuverlässige Ergebnisse erzielt.

3. Der Nutzer: Der Nutzer interagiert mit dem Assistenzsystem, um Empfehlungen oder Lösungen für spezifische Probleme zu erhalten. Hierbei stellt der Nutzer Eingabeinformationen (engl. Input Information) zur Verfügung, während er seinerseits Ausgabeinformationen (engl. Output Information) in Form von Empfehlungen, Lösungen oder Erklärungen erhält. Die Bandbreite der Nutzer reicht von Experten auf dem Gebiet bis hin zu Personen mit begrenztem Fachwissen.

Wissensbasierte Assistenzsysteme finden Anwendung in vielfältigen Bereichen, darunter medizinische Diagnose und Therapieplanung, Finanzanalyse sowie die Steuerung von Fertigungsprozessen [67]. Sie ermöglichen die Automatisierung komplexer Entscheidungsprozesse, tragen zur Fehlervermeidung bei und verbessern die Effizienz von Arbeitsabläufen. Aus diesen Gründen bieten sich wissensbasierte Assistenzsysteme optimal für die Konzeption eines sicheren und effektiven Systems zur Unterstützung des Anwenders bei Entscheidungen bezüglich der Bereitstellung von Technologiedaten an.

2.4 Sicherheit in der Informationstechnik

Die digitale, vernetzte Welt eröffnet einerseits sehr interessante Perspektiven für zukünftige industrielle IT-Anwendungen und Geschäftsmodelle. Andererseits ergeben sich durch die Digitalisierung erhebliche Sicherheitsrisiken. Folglich bestehen immer höhere Ansprüche an die Sicherheit sowohl im Office-Bereich „Information Technology (IT)“ als auch auf der Produktionsebene „Operation Technology (OT)“. Hierbei muss zudem das Zusammenspiel zwischen der Sicherheit und der Funktionalität eines Systems im Auge behalten werden [68].

Um den Sicherheitsanforderungen während der Konzeption zu entsprechen, ist es notwendig, die IT-Sicherheit mit Blick auf die diesbezüglich gesteckten Ziele zu analysieren. Das *National Institute of Standards and Technology (NIST)* stellt mit seiner *NIST SP 800-160*-Richtlinie eine Anleitung zur Einrichtung einer Systemsicherheitstechnik zur Verfügung, die analog zum Systemlebenszyklus entwickelt wird. Die Maßnahmen zur Entwicklung des Systems werden hierbei um die Sicherheitsmaßnahmen ergänzt. Die Sicherheitsmaßnahmen werden also parallel zu den Entwicklungsschritten implementiert [69].

2.4.1 Schutzziele

Schutzziele bezeichnen Anforderungen an ein System, die zum Schutz von wertvollen Gütern erfüllt werden müssen. Sie sind notwendig, um die IT-Sicherheit im Kontext der Informations- und Kommunikationstechnik messbar und bewertbar zu machen. Die Definition der IT- Sicherheit gemäß *Bundesamt für Sicherheit in der Informationstechnik* (BSI) stellt die Bedeutung der Hauptschutzziele in der IT-Sicherheit wie folgt dar:

> ***IT-Sicherheit ist der Zustand, in dem Vertraulichkeit, Integrität und Verfügbarkeit von Informationen und Informationstechnik durch angemessene Maßnahmen geschützt sind [70].***

Für eine sichere und vertrauenswürdige Funktionsweise eines Systems sind je nach Einsatzbereich und Anwendung einige Schutzziele in der IT-Sicherheit von Bedeutung, die in der Folge vorgestellt werden.

- ***Vertraulichkeit (engl. Confidentiality)***

Ein Gespräch zwischen zwei Personen gilt als vertraulich, wenn sichergestellt ist, dass die enthaltenen Informationen nicht an Dritte weitergegeben werden [71]. Um die Vertraulichkeit zu gewährleisten, müssen die Informationsflüsse zwischen zwei Parteien kontrolliert werden, sodass nur diese Parteien auf die ausgetauschten Informationen zugreifen können. So werden sowohl die Informationsinhalte als auch das Informationsverhalten diskret behandelt [72]. Die Vertraulichkeit wird gemäß DIN ISO/IEC 27000 folgendermaßen definiert:

> ***Vertraulichkeit ist die Eigenschaft, dass Informationen unberechtigten Personen, Einheiten oder Prozessen nicht verfügbar gemacht oder enthüllt werden [73].***

- ***Integrität (engl. Integrity)***

Die Integrität bezieht sich auf die Unversehrtheit, Unverfälschtheit und Korrektheit von Daten. Die Daten dürfen nicht durch Unbefugte erstellt oder manipuliert werden können [74] und [75]. Die DIN ISO/IEC 27000 definiert Integrität wie folgt:

> ***Integrität ist die Eigenschaft der Absicherung von Richtigkeit und Vollständigkeit von Werten [73].***

- ***Verfügbarkeit (engl. Availability)***

Das Schutzziel der Verfügbarkeit bezieht sich auf die Betriebsbereitschaft eines Systems. Verfügbar ist ein System, wenn dessen Funktionen zum Zeitpunkt der Nutzung bereitgestellt werden können und wenn diese Funktionen korrekt ablaufen [74] und [75]. Die Verfügbarkeit wird in der DIN ISO/IEC 27000 folgendermaßen definiert:

> ***Verfügbarkeit ist die Eigenschaft, einer berechtigten Einheit auf Verlangen zugänglich und nutzbar zu sein [73].***

- ***Authentizität (engl. Authenticity)***

Das Schutzziel der Authentizität schließt die Integrität mit ein. Es wird also nicht nur sichergestellt, dass der Inhalt einer Nachricht nicht verfälscht werden kann, vielmehr betrifft die Authentizität auch noch die Herkunft der Nachricht, also die Frage, ob der Sender einer Nachricht auch der Ersteller des Inhaltes ist [74] und [75]. „Authentizität“ wird ebenfalls in der DIN ISO/IEC 27000 definiert:

> ***Authentizität ist die Eigenschaft einer Einheit, das zu sein, was sie zu sein vorgibt [73].***

In diesem Zusammenhang werden oftmals zwei Begrifflichkeiten genannt: Je nach Perspektive in der Informationssicherheit ist von Authentifizierung oder Authentisierung die Rede. Eine überprüfte Person oder ein überprüftes System weist durch Authentisierung seine Identität nach. Das geschieht zum Beispiel durch eine Anmeldung mit dem Nutzernamen und einem Passwort oder dem Fingerabdruck. Nach dem Authentisierungsprozess erfolgt der

Authentifizierungsprozess. Das prüfende System stellt hierbei sicher, dass die angegebenen Daten der behaupteten Identität zugeordnet sind [76].

- ***Verbindlichkeit/ Nicht-Abstreitbarkeit (engl. Non-Reputability)***

Die Verbindlichkeit in der Kommunikation über das Internet wird anhand einer an einen Kommunikationspartner versendeten Nachricht sichergestellt. Die Verbindlichkeit einer Nachricht bezieht sich auf die Nicht- Abstreitbarkeit, es kann also auch gegenüber Dritten eindeutig nachgewiesen werden, wer der Autor der Nachricht war [74] und [75]. Nicht- Abstreitbarkeit wird in der DIN ISO/IEC 27000 wie folgt definiert:

> ***Nicht- Abstreitbarkeit ist die Fähigkeit, das Auftreten eines behaupteten Ereignisses oder einer Handlung und die verursachenden Einheiten nachzuweisen, um Streitigkeiten über das Auftreten oder Nichtauftreten des Ereignisses oder der Handlung und die Beteiligung von Einheiten an dem Ereignis zu entscheiden [73].***

- ***Autorisierung (engl. Authorization)***

Die Autorisierung fasst alle Nutzungsregeln von bereitgestellten Diensten zusammen und definiert die Zugriffsrechte einzelner Nutzer auf geschützte Ressourcen. Hierbei wird nach Überprüfung der festgelegten Regeln der Zugriff auf Dienste oder Daten gewährt oder verweigert. Die Autorisierung ist also eine organisatorische Maßnahme bezüglich der Vergabe von Zugriffsberechtigungen und Zugangsrechten von Subjekten auf bestimmte Objekte. Beispielsweise darf ein Nutzer eine Datei lesen, ändern oder löschen, während ein anderer Nutzer sie nur lesen darf [75]. Im Rahmen der Kommunikation zwischen zwei Teilnehmern entsteht ein Informationsfluss. Autorisierung bedeutet, dass ein Teilnehmer berechtigt ist, auf bestimmte Informationsinhalte zuzugreifen. Dafür muss dieser Teilnehmer zunächst authentifiziert werden, es muss also die Identität des Teilnehmers überprüft werden [74] und [77].

Die Schutzziele in der IT-Sicherheit sind essenziell für die entwickelte Konzeption zur Unterstützung des Anwenders bei der Nutzung von Technologiedatenmarktplätzen. Daher werden sie in dieser Dissertation im Rahmen der Erstellung des Anforderungsprofils, der Konzeptentwicklung und schließlich der Implementierung in den Blick genommen.

2.4.2 Technische Schutzmaßnahmen

Die Kryptografie ist die Wissenschaft von der Verschlüsselung von Daten mittels verschiedener Methoden [77] und [78]. Durch technologischen Fortschritt und die zunehmende Vernetzung und Globalisierung nehmen die Sicherheitsrisiken kontinuierlich zu. Daher kommt der Kryptografie große Bedeutung hinsichtlich des Schutzes von Informationen zu [79] und [80]. Ein kryptografisches System (kurz: „Kryptosystem“) besteht aus Klartexten (M), Schlüsseln (K), Geheimtexten (C) und Algorithmen zur Chiffrierung (E) und zur Dechiffrierung (D). Chiffrierung (engl. Encryption) und Dechiffrierung (engl. Decryption) werden wie folgt beschrieben: $E_k(M) = C$ und $D_k(C) = M$, wobei gilt: $D_k(E_k(M)) = M$ [81].

Für den Datenaustausch im Rahmen des Laserbearbeitungsprozesses ist eine Ende-zu-Ende-Sicherheit der Daten notwendig. Das bedeutet, dass Daten als Kryptotexte verschlüsselt vom Absender zum Empfänger übertragen werden. Die Schutzziele sollen durchgängig bis zur Datennutzung an den Maschinen gewährleistet sein. In dieser Dissertation wird die Kryptografie als wichtiges Werkzeug zur Konzeption der Datensicherheit angesehen. Deshalb werden im Folgenden die wichtigsten Kryptografie- Methoden diskutiert und dargestellt.

Symmetrische Verschlüsselung

Symmetrische Verschlüsselungsalgorithmen, auch als Secret-Key-Verschlüsselung bekannt, sind dadurch gekennzeichnet, dass es nur einen Schlüssel zur Decodierung und Verschlüsselung gibt. Dieser geheime Schlüssel muss also dem Absender und dem Empfänger bekannt sein. Er muss im Voraus auf sichere Weise ausgetauscht und geheim gehalten werden [78]. Das Prinzip des symmetrischen Verschlüsselungsverfahrens wird in der Abbildung 2-8 dargestellt.

Der gängigste symmetrische Algorithmus ist heutzutage das AES-Kryptosystem (engl. Advanced Encryption Standard). Es zeichnet sich durch eine einfache Implementierung, einen hohen Sicherheitsstandard und eine zeitsparende Anwendung aus [74]. Der AES-Algorithmus wurde 2001 vom *National Institute of Standards and Technology (NIST)* genormt [82]. Im Rahmen dieses Verfahrens werden Blöcke mit einer festen Größe von 128 Bits mit Hilfe von Schlüsseln von 128, 192 und 256 Bits codiert. Je nach Schlüssellänge werden diese Algorithmen als AES-128, AES-192 oder AES-256 bezeichnet [83].

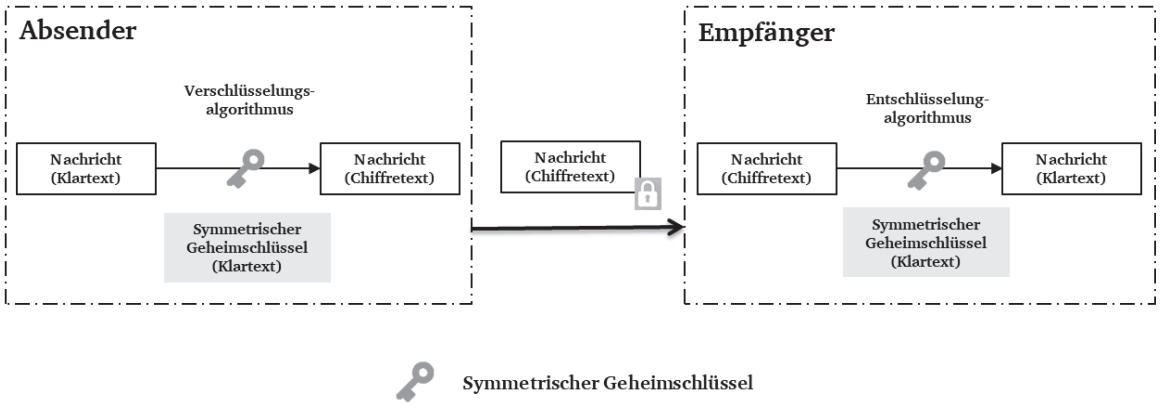

Abbildung 2-8: Die symmetrische Verschlüsselung (eigene Darstellung)

Asymmetrische Verschlüsselung

Bei der asymmetrischen Verschlüsselung, auch Public-Key-Verschlüsselung genannt, wird ein Schlüsselpaar zur Ver- und Entschlüsselung des Chiffretexts benötigt. Dieses Paar besteht aus einem öffentlichen und einem privaten Schlüssel. Mit dem öffentlichen Schlüssel des Empfängers kann der Absender eine verschlüsselte Nachricht senden. Die Decodierung der verschlüsselten Nachricht durch den Empfänger ist jedoch nur über den privaten Schlüssel des Letzteren möglich. Der Absender muss hingegen keinen eigenen Schlüssel besitzen, um für einen Empfänger eine Nachricht zu verschlüsseln und an ihn zu schicken [84]. Die Abbildung 2-9 stellt das Prinzip des asymmetrischen Verschlüsselungsverfahrens dar.

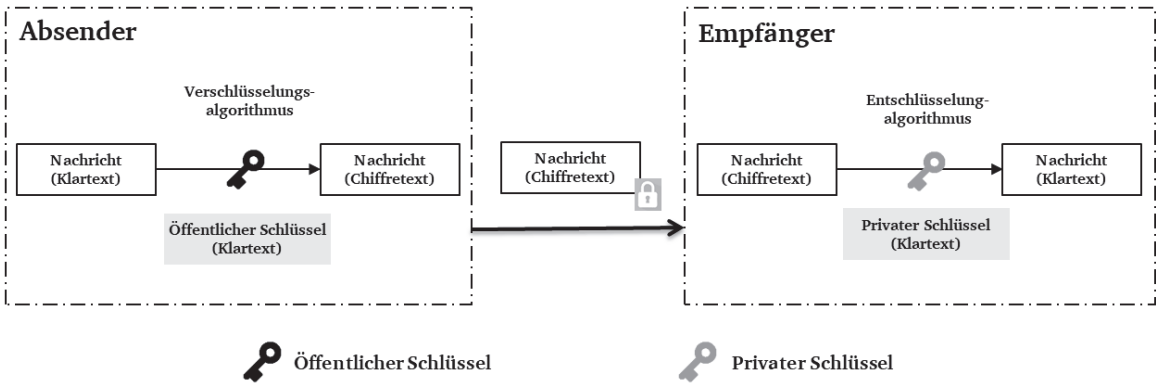

Abbildung 2-9: Die asymmetrische Verschlüsselung (eigene Darstellung)

Die Vorteile der asymmetrischen Verschlüsselung bestehen hauptsächlich darin, dass kein geheimer Schlüssel ausgetauscht werden muss. Allerdings ist dieses Verfahren ineffizient im Vergleich zum symmetrischen Verfahren und es bedarf eines Schlüsselmanagements der

Kommunikationsteilnehmer. Die Authentizität der Schlüssel wird mittels Zertifikate sichergestellt [74]. Im Vergleich zu symmetrischen Algorithmen sind asymmetrische Algorithmen langsamer und benötigen längere Schlüssel, um das gleiche Maß an Sicherheit zu gewährleisten [85].

Das RSA-Verfahren [177] wurde von Ronald Rivest, Adi Shamir und Leonard Adleman entwickelt. Der RSA-Algorithmus ist bis heute das asymmetrische Verfahren, das am einfachsten zu implementieren ist. Hierfür wird eine Schlüssellänge von mindestens 2048 Bits empfohlen. Der RSA-Algorithmus wird häufig zur Verschlüsselung von Dateien, zum sicheren Austausch von geheimen Schlüsseln und zur Überprüfung von digitalen Signaturen eingesetzt [74] und [81].

Kryptografische Hashfunktionen

Kryptografische Hashfunktionen werden häufig zur Überprüfung der Integrität von Daten verwendet. Dabei werden einmalige digitale Fingerabdrücke der Daten errechnet und zusammen mit diesen versandt. Der Empfänger berechnet zunächst den Hashwert der Daten und vergleicht diesen mit dem assoziierten Fingerabdruck. Sind diese Werte identisch, wird davon ausgegangen, dass die Daten nicht manipuliert oder modifiziert wurden [74]. Sichere kryptografische Hashfunktionen müssen kollisionsresistent sein und es muss sich um Einwegfunktionen handeln. Dadurch kann die Integrität der Daten sichergestellt werden [77].

Durch die Hashfunktion werden Nachrichten beliebiger Länge auf Nachrichten einer festen Länge, üblicherweise sind das 256 Bit, komprimiert. Hierfür wird am häufigsten das Verfahren der SHA-Familie (engl.: Secure Hash Algorithm) angewandt. Hashfunktionen werden oftmals in Kombination mit digitalen Signaturverfahren eingesetzt, um die Integrität der Daten zu prüfen und die Datenherkunft nachzuweisen. Hierbei wird der Hashwert der Daten berechnet und signiert, anstatt die gesamten Daten zu signieren. Damit wird der Berechnungsaufwand wesentlich reduziert [77] und [78].

Digitale Signaturen

Zum Erstellen einer digitalen Signatur werden kryptografische Hashfunktionen und asymmetrische Verschlüsselungsverfahren kombiniert. Mit Hilfe der digitalen Signaturen können die Integrität der Daten sowie die Echtheit der Datenherkunft sichergestellt werden [77].

Der Absender berechnet den Hashwert für seine Nachricht anhand einer kryptografischen Hashfunktion, z. B. SHA-2. Danach verwendet er seinen privaten Schlüssel für die Verschlüsselung des berechneten Hashwertes. Das Ergebnis ist dann die digitale Signatur. Schließlich schickt der Absender seine Nachricht zusammen mit der digitalen Signatur an den Empfänger. Der Empfänger nutzt dann den ihm bekannten öffentlichen Schlüssel des Absenders zum Decodieren des verschlüsselten Hashwerts und berechnet den Hashwert der empfangenen Nachricht. Zur Prüfung vergleicht der Empfänger schließlich den selbst errechneten Haschwert mit den entschlüsselten Hashwert aus der Signatur. Stimmen die Hashwerte überein, ist die digitale Signatur des Absenders verifiziert. Nun wurde dem Empfänger nachgewiesen, dass die Inhalte der Nachricht korrekt sind und vom gewünschten Absender stammen. Wenn die Nachricht nach dem Senden manipuliert wurde, sind die Hashwerte nicht identisch [74], [77] und [78].

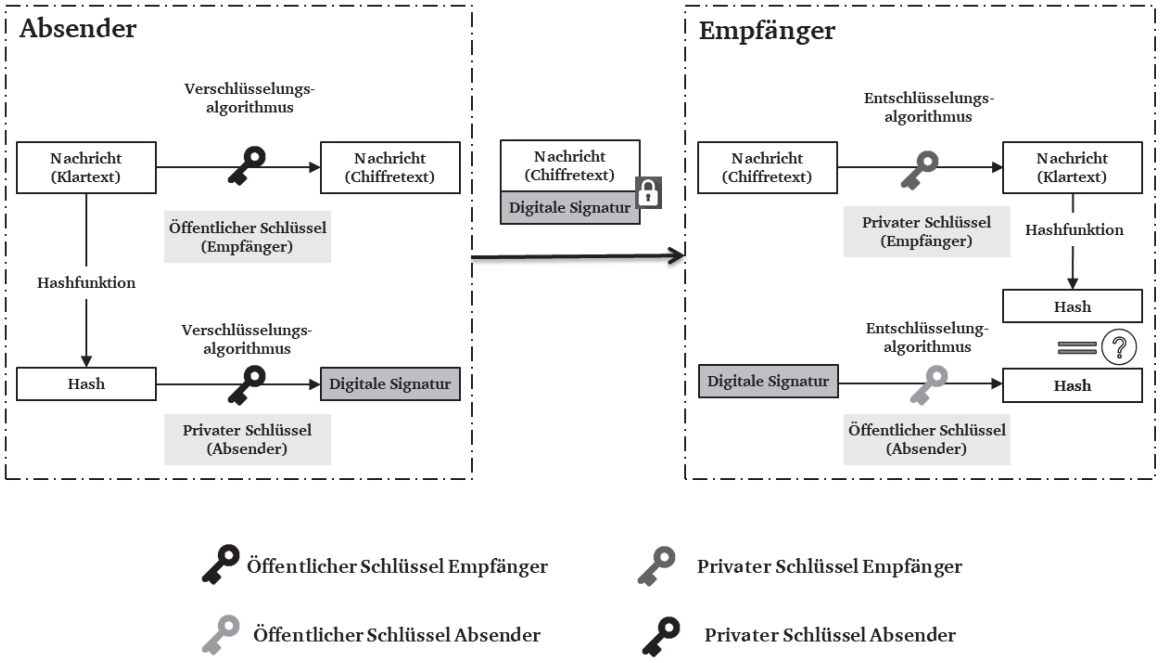

Abbildung 2-10: Digitale Signaturen (eigene Darstellung)

Für die sichere Verifizierung der digitalen Signaturen kommen sowohl kollisionsresistente kryptografische Hashfunktionen als auch asymmetrische Verschlüsselungsverfahren zum Einsatz.

Zertifikate

Zertifikate sind so etwas Ähnliches wie elektronische Personalausweise für Personen und Systeme. Sie ermöglichen eine sichere und authentische Kommunikation über das Internet. Sie werden für die Verwaltung von öffentlichen Schlüsseln, die für kryptografische Technologien benötigt werden, sowie für deren Zuordnung zu bestimmten Teilnehmern verwendet. Zertifikate dienen der Vertrauensbildung, indem sie die Authentizität der Personen und Systeme garantieren. Sie werden normalerweise von einer Zertifizierungsstelle ausgestellt, verwaltet und signiert, zum Beispiel von der *Public-Key-Infrastruktur* (PKI). Des Weiteren beinhalten sie wichtige Informationen zum Teilnehmer ID(A), zu seinem öffentlichen Schlüssel e_A, zur Gültigkeit und zur Zertifizierungsstelle und deren Signatur Sig_T [78].

$$Z(A) = (\mathrm{ID(A)}, e_A, Sig_T(ID(A), e_A))$$

Die heutzutage am häufigsten verwendeten Zertifikate werden nach dem Standard *X.509* aufgebaut [86].

2.4.3 Blockchain-Technologie

Die Blockchain-Technologie ist eine netzwerkbasierte dezentrale Technologie, die Informationen sicher und transparent kommunizieren und übertragen kann [87], [88]. Eine Blockchain besteht aus einer Kette von Blöcken, die kryptografisch miteinander verknüpft sind und Informationen über durchgeführte Transaktionen enthalten. Die Blöcke werden durch einen Konsensmechanismus gesichert, der die Gültigkeit der Transaktionen bestimmt. Jeder Block ist dabei durch einen kryptografischen Hashwert mit dem offiziellen Block verknüpft. Diese Informationen sind Gegenstand eines wirkungsvollen Kontrollmechanismus, wodurch die Integrität der Daten sichergestellt wird [89]. Durch die Kombination von Kryptografie, gemeinsamem Konsens und Dezentralisierung bietet die Blockchain-Technologie ein hohes Maß an Sicherheit, Integrität und Transparenz. Blockchain-Systeme können in verschiedenen Ausprägungen konzipiert werden. Zwei wesentliche Faktoren beeinflussen ihre Gestaltung: der Grad der Zentralisierung und die Erfordernis einer Genehmigung zur Teilnahme am Verwaltungsprozess des Blockchain-Netzwerks [90].

Die Blockchain-Technologie hat das Potenzial, Anwendungen in verschiedenen Bereichen zu revolutionieren, darunter Finanzen, Transport und Gesundheitswesen. Die Technologie kann auch zur Überprüfung der Authentifizierung von Dokumenten und Identitäten eingesetzt

werden [91] und [92]. Sie bietet im Vergleich zu stark zentralisierten Datenbanken ein höheres Maß an Sicherheit, Transparenz und Datenschutz. Die Integration der Blockchain-Technologie dient der Effizienzsteigerung, der Vermeidung von Manipulationen und der Erhöhung der Transparenz [93], [94] und [95].

2.4.4 Smart Contracts

Ein Smart Contract ist ein digitaler Vertrag, der auf der Blockchain-Technologie basiert. Er wird auf automatisierte Weise erstellt, verwaltet und ratifiziert. Dadurch werden manipulationssichere und transparente Transaktionen automatisch durchgeführt und aufgezeichnet. Die intelligenten Verträge werden nur dann durchgeführt, wenn bestimmte Bedingungen erfüllt sind. Diese Bedingungen werden zwischen den Vertragsparteien vereinbart und auf der Blockchain gespeichert und können nicht nachträglich geändert werden [96]. Somit ermöglichen intelligente Verträge ein effizienteres, transparenteres und sichereres Vertragsmanagement.

Smart Contracts werden in dezentralen Anwendungen eingesetzt und ermöglichen eine automatisierte Abwicklung digitaler Verträge ohne Zwischenhändler. Sie können auch die Überprüfung digitaler Identitäten unterstützen, indem sie an der Erstellung von digitalen Identitätsnachweisen beteiligt sind. Digitale Verträge haben ein großes Potenzial, verschiedene Branchen zu revolutionieren und in Bereichen wie Finanzdienstleistungen, Lieferkettenmanagement und der digitalen Identitätsverifizierung Einzug zu halten [97].

Es gibt immer Herausforderungen hinsichtlich Skalierung, Sicherheit und rechtliche Durchsetzbarkeit, die bedacht werden müssen [98] und [99]. Wird diese Technologie weiterentwickelt, hat sie das Potenzial, durch erhöhte Sicherheit und Transparenz den Modus der Abwicklung digitaler Verträge zu revolutionieren.

2.4.5 Zugriffsberechtigung auf Daten und Lizenzierung

Der Zugriff auf die Daten kann auf verschiedene Arten reguliert werden. Beispielsweise kann er an die Identität einer Person gebunden sein, der bestimmte Rechte eingeräumt werden [74]. Darüber hinaus existieren rechtliche und technische Lösungen, um die Zugriffsrechte einer Entität auf spezifische Inhalte zu steuern. Die Berechtigungen können dabei in Form von rollenbasierten, identitätsbasierten oder attributbasierten Modellen implementiert werden.

Die Autoren von [100] befassen sich mit dem Thema der Datenhoheit, insbesondere in Bezug auf den Zugriff auf Nutzungsdaten von Werkzeugmaschinen und deren Zuordnung zu diesen Maschinen. In ihrer Veröffentlichung entwickeln sie eine Checkliste für die Bestandteile eines Vertrags, der die Zugriffs- und Nutzungsrechte in Bezug auf solche Daten regelt. Dieser Vertrag enthält Informationen über die Vertragsparteien, den Vertragsgegenstand, die Leistungspflichten, Zugriffsrechte und Nutzungszwecke, die Beschaffenheit des Vertragsgegenstands, die Vertragslaufzeit, Kündigungsmöglichkeiten, die Haftung für Schäden durch fehlerhafte Daten, Sanktionsmöglichkeiten bei Vertragsverletzungen sowie idealerweise Definitionen der verwendeten Begriffe, Erläuterungen zur technischen Umsetzung und den eingesetzten Systemen sowie kurze Hintergrundinformationen zum Vertrag selbst [101].

Zusätzlich zur Datenbeschaffenheit sollte der Vertrag auch die Datenbereitstellung regeln. Dies umfasst beispielsweise die Festlegung der zulässigen Schnittstellen und gewünschten Datenformate. In diesem Zusammenhang sind die verwendeten Technologien und technischen Details von großer Bedeutung [100]. Solche Verträge können mithilfe eines Lizenzierungssystems umgesetzt werden, das die Zugriffsrechte auf die Daten reglementiert. In der Vermarktung von digitalen Daten sind Lizenzverträge mittlerweile weit verbreitet. Beispielsweise werden Lizenzverträge in der Musikindustrie [102] und bei kommerzieller Software häufig eingesetzt [103]–[106].

Die Lizenzierung stellt eine Möglichkeit dar, geistiges Eigentum (engl. Intellectual Property) zu vermarkten. Dafür muss eine Lizenzvereinbarung zwischen dem Inhaber des geistigen Eigentums, auch „Lizenzgeber“ genannt (engl. Licensor), und dem Nutzer, genannt „Lizenznehmer“ (engl. Licensee), getroffen werden. Mit dieser Vereinbarung wird dem Lizenznehmer die Berechtigung erteilt, das geistige Eigentum des Lizenzgebers unter den im Vertrag festgelegten Bedingungen und unter Rücksichtnahme auf das Urheberrecht zu nutzen [107].

Das bedeutet, dass die Lizenzvereinbarung dazu dient, das geistige Eigentum und das Urheberwissen auf der Grundlage der Urheberrechtsgesetze zu schützen. Die Lizenz legt auch die Verantwortlichkeiten der Parteien fest, die in der Lizenzvereinbarung schriftlich festgehalten werden. Moderne Lizenzierungen erfolgen nicht mehr ausschließlich auf traditionelle Weise mit schriftlichen Vereinbarungen, sondern werden oft mithilfe von Softwaretools und in elektronischer Form umgesetzt [108].

Gemäß *Bürkner* ist der Ausdruck „Lizenzierung“ ein Sammelbegriff für Maßnahmen, um Software gegen illegale Nutzung zu schützen und um flexible Vermarktungsmodelle zu realisieren [108]. Im Duden wird der Begriff „Lizenz“ als „[g]egen eine Gebühr erteilte rechtskräftige Genehmigung“ definiert. Hier werden „Erlaubnis“, „Berechtigung“, „Befugnis“ und „Ermächtigung“ als Synonyme für den Begriff „Lizenz“ angeführt [109].

Die Lizenzierung bringt viele wirtschaftliche Vorteile mit sich, sie eröffnet neue Erwerbsmöglichkeiten und Märkte und ermöglicht die Umsetzung neuartiger Geschäftsmodelle [107]. Zum Beispiel können Unternehmen ihr innovatives Fachwissen und ihre Technologien kleinen und mittleren Unternehmen durch Lizenzverträge zur Verfügung stellen [128] und dabei Profite generieren [110].

Lizenzformen

Die am häufigsten mit Lizenzen vermarkteten Softwareprodukte sind die proprietäre und die freie Software. Die Nutzung einer proprietären Software wird vertraglich zum Beispiel über eine sogenannte „Endbenutzer-Lizenzvereinbarung“, abgekürzt als EULA (engl. End User License Agreement) reglementiert. Dabei hat der Nutzer keinen Zugriff auf den Quelltext.

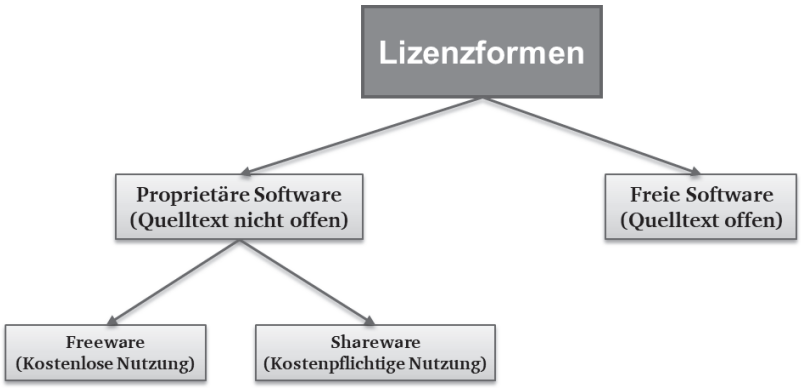

Abbildung 2-11: Lizenzformen (eigene Darstellung)

Freie Software (auch: Open Source) kann vom Nutzer zu jedem Zweck verwendet, angepasst und weitergegeben werden. Beispiele dafür sind das Betriebssystem *Linux* und das E-Mail-Programm *Thunderbird*. Hierbei ist ein Zugriff auf den Quelltext erlaubt. Die Lizenzvereinbarung GPL (engl. General Public Licence) wird häufig für freie Software verwendet [103].

Bei Proprietärer Software wird zwischen Freeware und Shareware unterschieden. Die Nutzung von Freeware ist zwar kostenlos, sie wird aber durch Endbenutzer-Lizenzvereinbarungen beschränkt. Ein Beispiel dafür ist der *Adobe Acrobat Reader*. Shareware kann über einen gewissen Zeitraum hinweg kostenlos für bestimmte Zwecke benutzt werden. Danach fallen Gebühren an oder die Software ist nur noch eingeschränkt nutzbar [103]. Die obige Abbildung 2-11 stellt die Lizenzformen dar.

Lizenzmodelle

Grundsätzlich gibt es zwei Typen von Lizenzen: statische und dynamische Lizenzen. Statische Lizenzen sind an ein System oder einen Anwender gebunden. Deren Übertragung ist nur in Absprache mit dem Lizenzgeber möglich. Dynamische Lizenzen können zum Bedarfszeitpunkt flexibel entweder ad hoc angefordert werden oder es besteht ein Kontingent an Nutzungen, das sukzessive aufgebraucht wird [111]. Eine Lizenz wird nach dem erfolgreichen Erwerb je nach vereinbartem Kaufmodell und je nach Kaufbedingungen individuell generiert. Mit verschiedenen Lizenzmodellen gehen unterschiedliche Berechtigungsrichtlinien einher. Zum Beispiel gibt es Lizenzmodelle für eine einmalige Nutzung, für Pay- per- Use, für Festpreispakete oder für langfristige Abonnements.

In [63] werden drei mögliche Lizenzmodelle definiert, mittels derer Technologiedaten auf entsprechenden Marktplätzen bezogen werden können: Es gibt Basislizenzmodelle für Pay- per- Use sowie zeitbasierte und nutzungsunbegrenzte Lizenzen. Von diesen Lizenzmodellen lassen sich weitere Lizenzmodellen ableiten.

Der Einsatz von neuen Technologien erfordert eine dynamische und umfangreiche Lizenzmodellierung, die sämtliche Nutzerbedürfnisse abdeckt. Für die Erzeugung einer Lizenz muss ein Nutzer das Lizenzmodell basierend auf seinen Bedürfnisse und seiner IT- Architektur sowie hinsichtlich des geplanten Einsatzes auswählen. Für ein Lizenzmodell sind normalerweise *Lizenzart*, *Lizenztyp*, *Lizenzklasse* und *Lizenzmetrik* bestimmend [103]. Auf untenstehender Abbildung sind die gängigsten Lizenzmodelle abgebildet.

Es gibt zwei Hauptarten von Lizenzen: Einzel- und Mehrplatzlizenzen. Eine Einzelplatzlizenz ist an ein System gebunden [111]. Bei Mehrplatzlizenzen kann die Software von mehreren Systemen verwendet werden [103]. Mittels einer Paketlizenz (engl. Packages) wird die Nutzung mehrerer voneinander unabhängiger Produkte oder Dienste ermöglicht, statt einzelne Lizenzen für jedes Produkt oder jeden Dienst zu erwerben [108]. Mittels Gruppierungslizenzen können

für Nutzergruppen oder für mehrere Systeme Nutzungsrechte erworben werden. Zum Beispiel kann ein Nutzer das lizenzierte Produkt auf mehreren Systemen gleichzeitig nutzen, was dann als eine einzige Nutzung gewertet wird [108].

Mit der Lizenzklasse sind Informationen zu Versionen, Updates, Upgrades und Leistungsmerkmalen eines Produktes verknüpft [103]. Der Lizenztyp definiert die Bindung einer Lizenz an ein System, einen Nutzer oder einen Standort. Wird zum Beispiel eine nutzergebundene Lizenz erteilt, muss die Identität der Nutzer vor der Nutzung des Produktes überprüft und nachgewiesen werden. Genauso muss die Kennung eines Hardwaresystems oder eines Standortes bei hardware- und standortgebundenen Lizenzen vor der Nutzung nachgewiesen werden. Floating-Lizenzen (engl. Concurrent Licences), auch Netzwerklizenzen genannt, sind hingegen nicht auf ein einziges System beschränkt, sondern erlauben die Nutzung des Produktes durch einen beliebigen Nutzer auf mehreren unterschiedlichen Systemen, die in ein Netzwerk integriert wurden [103], [108], [111] und [112].

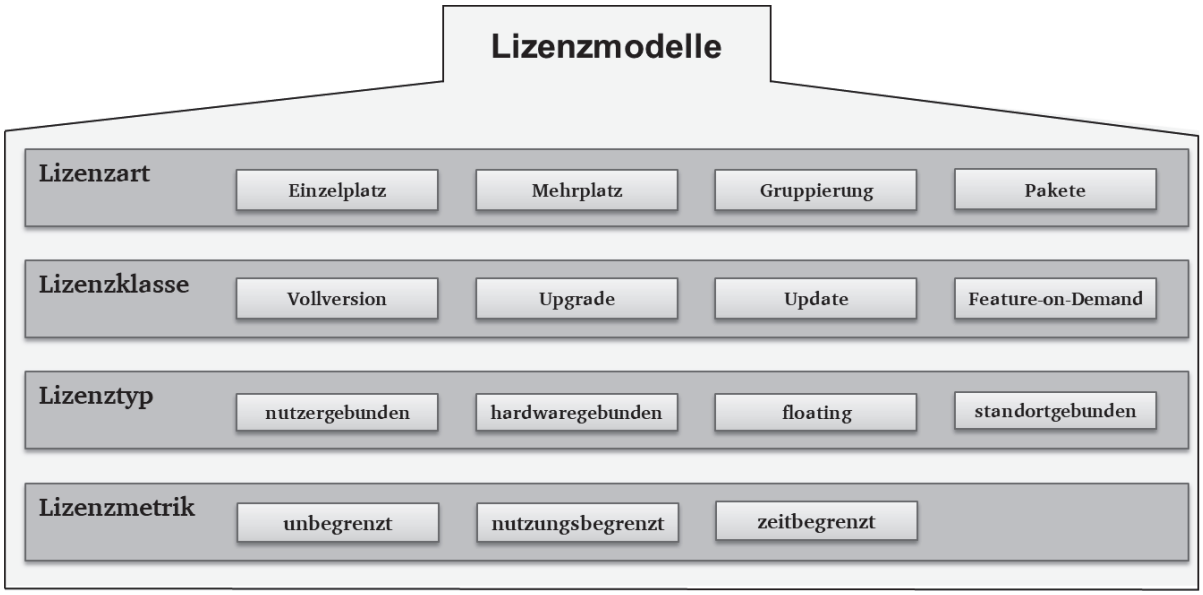

Abbildung 2-12: Übersicht über Softwarelizenzmodelle (eigene Darstellung angelehnt an: [103], [104], [106], [108] und [111]–[113])

Bei der systemgebundenen Lizenzierung wird ein „digitaler Fingerabdruck“ des Systems generiert, der vom Lizenzgeber validiert werden muss. Nach der Validierung und Überprüfung verschickt der Lizenzgeber automatisch eine verschlüsselte und signierte Lizenz- bzw. Aktivierungsdatei, die nur zu diesem einen Gerät passt [104]. Während Lizenzart, Lizenzklasse und Lizenztyp der Klassifizierung einer Lizenz dienen, beschreibt die Lizenzmetrik die technischen Nutzungsbedingungen einer Lizenz [103]. Bei nutzungsbegrenzten Lizenzen sind

die Metriken zum Beispiel an die Kapazität eines Systems, an das Datenvolumen oder an die Häufigkeit der Nutzung gebunden. Das gebräuchlichste Lizenzmodell ist Pay- per- Use, auch On-Demand-Lizenz genannt [111]. Hierbei wird nur die tatsächliche Nutzung des Produktes nachträglich in Rechnung gestellt. Dagegen erlaubt eine nutzungsunbegrenzte Lizenz, wie der Name bereits verrät, die unlimitierte Nutzung des Produkts. Zeitbegrenzte Lizenzen sind nach dem Ablauf einer vorweg festgelegten Zeitspanne nicht mehr gültig [103], [108] und [112]. Mit den oben dargestellten Merkmalen von Lizenzmodellen wird kein Anspruch auf Vollständigkeit erhoben, da stetig neue Modelle den sich laufend wandelnden Marktbedürfnissen entsprechend entwickelt werden.

Auslieferung von erworbenen Lizenzen

Lizenzen können als Datei auf einer Hardwarekomponente oder als Softwaredatei über das Internet an den Kunden ausgeliefert werden. Bei der ersten Variante liefert der Lizenzgeber die Lizenz auf einer externen Hardwarekomponente, dem sogenannten „Dongle“ – das kann beispielsweise ein USB- Stick oder eine Micro- SD-Karte sein. Das Dongle muss mit einer passenden Systemschnittstelle verbunden werden, damit die Lizenzinformationen abgefragt werden können. In der Folge wird die Nutzung freigegeben oder eventuell unterbunden. Hierbei wird der gespeicherte Hardware- Code ausgelesen, der angibt, ob eine gültige Lizenz vorliegt. Ohne das Anschließen des Dongles an die Systemschnittstelle können die Daten nicht verwendet werden. Das bedeutet, dass die gebundene Lizenz nicht mehr gültig ist, wenn das Dongle ausfällt. In diesem Fall muss eine neue Lizenz beim Lizenzgeber angefragt werden.

Nun werden die Vor- und Nachteile der hardwarebasierten Lizenzierung dargestellt. Ein großer Vorteil hierbei ist die Mobilität und Flexibilität der Lizenzen. Ein Dongle kann Technologiedaten auf unterschiedlichen Maschinen freigeben. Außerdem bedarf es für die Nutzung keines Internetzugangs. Allerdings ist der Umgang mit Dongles aufwendig und kompliziert, und es fallen Logistikkosten sowie ein erheblicher Verwaltungsaufwand bei den Anwendern an. Zudem besteht die Gefahr, dass ein Dongle verloren geht und somit die Technologiedaten nur verspätet genutzt werden können, womit eventuell die Abarbeitung von Aufträgen verzögert wird.

Lizenzen können auch als Softwaredatei auf einem Webserver bereitgestellt und über das Internet heruntergeladen oder per E- Mail verschickt werden – die softwarebasierte Lizenzierung. Der Lizenznehmer „Anwender“ erhält dabei vom Lizenzgeber „Technologiedatenmarktplatz“ eine verschlüsselte und signierte Lizenzdatei, mittels derer die Daten freigeschalten werden können. Der Lizenznehmer meldet sich hierfür über das Internet

an und ruft die Lizenzen bedarfsgerecht ab. Es erfolgt demnach zunächst die Authentifizierung des Anwenders. Diese kann über die Eingabe eines zuvor gelieferten Codes oder von persönlichen Anmeldungsdaten erfolgen. Im Vergleich zur hardwarebasierten Lizenzierung entfallen bei der softwarebasierten Lizenzierung die Versandkosten sowie die Verwaltungsarbeit und es besteht keine Verlustgefahr. Für die Vermarktung von Technologiedaten werden auf den entsprechenden Online- Marktplätzen möglichst unterschiedliche Lizenz- und Preismodelle bereitgestellt. Dank dieser Vielfalt können die Anwender ihre Kosten und den Aufwand minimieren. So wird der Erwerb von Technologiedaten effektiver, flexibler und attraktiver. In diesem Sinn eignen sich softwarebasierte Lizenzierungen für das vorliegende Konzept besser als hardwarebasierte Vorgehensweisen, weshalb in der Folge auf Erstere zurückgegriffen wird.

2.4.6 Blockchain- und Smart-Contract-Technologien für die Lizenzierung digitaler Inhalte

Die Kombination von Blockchain- und Smart-Contract-Technologien kann zahlreiche Vorteile für die Lizenzierungsindustrie bringen. Zum Beispiel können Intermediäre entfallen, wodurch Transaktionskosten gesenkt und die Transparenz sowie die Nachverfolgbarkeit erhöht werden. Mit Hilfe von Smart Contracts können Lizenzvereinbarungen automatisch ausgeführt werden, was das Risiko von Betrug und Fehlern reduziert [114]. Dabei können Lizenznehmer und Lizenzgeber die Einhaltung der Vereinbarung auf der Blockchain jederzeit überwachen. Beispielsweise kann ein Smart Contract so programmiert werden, dass er die Nutzung eines lizenzierten digitalen Inhalts automatisch überwacht und die Lizenz deaktiviert, wenn die Nutzungsbedingungen nicht mehr erfüllt werden. Bei der Verwendung von Blockchain- Technologie für die Lizenzierung sind digitale Identitäten von großer Bedeutung. Jeder Teilnehmer im Lizenzierungsprozess kann eine eindeutige digitale Identität, zum Beispiel in Form eines Zertifikats, erhalten, die auf der Blockchain gespeichert wird. Dadurch wird das Vertrauen zwischen den Parteien gestärkt und die Transparenz und Sicherheit des Prozesses verbessert [115].

Der Autoren in [116] schlagen eine neue Plattform namens LUCE vor, die auf der Blockchain- Technologie basiert und die gemeinsame Nutzung von Daten sowie die Einhaltung von Lizenzvereinbarungen überwacht. Die Autoren argumentieren, dass aktuelle Plattformen zur gemeinsamen Nutzung von Daten in Bezug auf Transparenz, Rückverfolgbarkeit und Rechenschaftspflicht Mängel aufweisen, was potenzielle rechtliche und ethische Probleme

verursachen kann. Die LUCE-Plattform bietet einen sicheren und transparenten Mechanismus zur gemeinsamen Nutzung von Daten und zur Überwachung der Datennutzung. Sie nutzt ein blockchainbasiertes System zur Speicherung und Verwaltung von Datenlizenzen. Zudem werden intelligente Verträge eingesetzt, um die Überwachung und Durchsetzung der Lizenzbedingungen zu automatisieren. Die Plattform setzt sich aus einem Datenfreigabemodul, einem Lizenzverwaltungsmodul, einem Überwachungs- und Durchsetzungsmodul sowie einem Blockchain-Netzwerk zusammen. Das Datenfreigabemodul verwaltet die gemeinsame Nutzung von Daten durch verschiedene Nutzer. Es ermöglicht Daten hochzuladen und den Zugriff darauf zu erlauben, wobei sie die Nutzungsbedingungen, die Zugriffsberechtigungen sowie die Gebühren für jeden Lizenztypen festlegen. Datenkonsumenten können nach Daten suchen und den Zugriff darauf beantragen. Sobald der Zugriff gewährt wird, erfolgt ein verschlüsselter und sicherer Austausch der Daten zwischen den beiden Parteien. Das Lizenzmanagementmodul verwaltet die Zugriffsberechtigungen, indem es die Lizenzen auf dem Blockchain-Netzwerk speichert und deren Nutzung im Laufe der Zeit verfolgt. Es stellt sicher, dass die Datenkonsumenten die Lizenzbedingungen, wie beispielsweise die Dauer der Nutzung, die erlaubten Nutzungen und die Gebühren, einhalten. Wenn ein Verstoß festgestellt wird, kann das Modul automatisch rechtliche Maßnahmen einleiten. Das Überwachungs- und Durchsetzungsmodul überwacht die Datennutzung und setzt die Lizenzanforderungen durch. Hierbei werden Smart Contracts eingesetzt, um den Prozess zu automatisieren und manuelle Eingriffe zu reduzieren. Das Modul kann die Datennutzung in Echtzeit verfolgen und bei nicht autorisierten Aktivitäten Maßnahmen auslösen. Das Blockchain-Netzwerk bildet das Fundament der LUCE-Plattform und ermöglicht eine sichere, dezentrale Speicherung und Verfolgung von Datenlizenzen und Zugriffsberechtigungen. Dadurch wird eine transparente und manipulationssichere Verwaltung der Datenlizenzen gewährleistet [116].

Die Veröffentlichung [117] schlägt vor, elektronische Gesundheitsdaten mithilfe von Blockchain- und Smart-Contract-Technologien zu sichern. Derzeitige zentralisierte Ansätze zur Verwaltung von Gesundheitsdaten sind laut den Autoren anfällig für Sicherheitslücken. Ein dezentralisierter Ansatz mit Blockchain und intelligenten Verträgen kann dagegen mehr Sicherheit und Datenschutz bieten. Das vorgestellte System nutzt intelligente Verträge zur Verwaltung des Zugriffs auf Gesundheitsdaten und ermöglicht es medizinischen Dienstleistern, die für die Versorgung benötigten Daten abzurufen. Gleichzeitig haben die Patienten die Kontrolle über den Zugriff auf ihre eigenen Daten, was die Datensicherheit und den Datenschutz gewährleistet. Allerdings müssen bei der Einführung eines solchen Systems Herausforderungen in Bezug auf Skalierbarkeit, Interoperabilität sowie regulatorische und rechtliche Maßnahmen berücksichtigt werden [117].

Der vorgeschlagene Ansatz in [118] verwendet eine semantische Darstellung der Lizenzbedingungen, die präzisere und standardisierte Lizenzvereinbarungen ermöglicht. Diese Lizenzbedingungen werden dann in einer Blockchain aufgezeichnet, womit ein dezentrales und sicheres System zur Verfolgung und Überprüfung des Eigentums und der Nutzung digitaler Inhalte geschaffen wird. Die Autoren sind der Ansicht, dass derzeitige Lizenzierungssysteme für digitale Inhalte oft undurchsichtig und verwirrend sind, was zu rechtlichen Streitigkeiten und anderen Problemen führen kann.

Im Rahmen des Forschungsprojekts KOSMoS wurde ein privates Blockchain- Toolkit für industrielle Anwendungen entwickelt. Ein spezifischer industrieller Anwendungsfall zur Überwachung und Verfolgung von Laserproduktionsprozessen wurde dabei behandelt. Im Rahmen des KOSMoS-Systems wurden intelligente Verträge genutzt, um Regeln für jeden Prozessschritt in der Laserproduktion festzulegen und um zu überprüfen, ob jeder Schritt erfolgreich abgeschlossen wurde, bevor mit dem nächsten fortgefahren wird. Ein weiterer wichtiger Bestandteil des KOSMoS-Systems ist die Verwendung von Sensoren zur Erfassung von Daten über den Produktionsprozess. Zum Beispiel können Temperatursensoren verwendet werden, um die Temperatur des Lasers während der Produktion zu überwachen und somit eine optimale Qualität des Endprodukts zu gewährleisten. Diese Daten werden in Echtzeit erfasst und auf der Blockchain gespeichert. Das KOSMoS-System verfügt auch über eine intuitive Benutzeroberfläche, die dem Nutzer den aktuellen Status des Produktionsprozesses sowie detaillierte Informationen über die verwendeten Materialien und Parameter in Echtzeit bereitstellt [119].

Es existieren bereits einige Plattformen, welche Blockchain- und Smart-Contract-Technologien erfolgreich in der Verwaltung von digitalen Lizenzen nutzen. Im Folgenden sind einige Beispiele aufgeführt:

1. *Monegraph* und *Verisart* sind Plattformen, die auf der Blockchain-Technologie basieren und digitale Kunstwerke lizenzieren. Künstler können ihre Werke sicher und transparent über die Plattformen verkaufen.

2. *Ujo Music* und *KODA* sind Plattformen, die Musikern die Möglichkeit bieten, ihre Musik direkt an Fans zu verkaufen und gleichzeitig die Kontrolle über ihre Musik und die Lizenzbedingungen zu behalten.

3. *Mediachain, KodakOne, IPStock* und *SingularDTV* sind Plattformen, auf denen Fotografen, Videografen und andere Kreative ihre digitalen Inhalte sicher und transparent verkaufen können.

4. *Lition* ist eine auf Blockchain-Technologie basierende Plattform, die Energie- und Handelsunternehmen Lösungen zur Verfügung stellt, um ihre Geschäftsprozesse effizienter und transparenter zu gestalten.

Der aktuelle Stand der Technik belegt die erfolgreiche Anwendung der Blockchain- und Smart-Contract-Technologien in verschiedenen Branchen, insbesondere in der Kunst und der Musik, im Sinne der Automatisierung des Lizenzierungsprozesses für digitale Inhalte. In einigen industriellen Branchen wird ebenfalls bereits in begrenztem Maße Blockchain-Technologie für die Überwachung und Lizenzierung von Maschinen eingesetzt [116] und [119]. Diese Beispiele bestätigen, dass diese Technologien die Entwicklung neuer Geschäftsmodelle in Industriebranchen, die auf Daten basieren, ermöglichen können.

Bisher sind keine bekannten Anwendungen der Blockchain-Technologie und intelligenter Verträge für die effiziente und maßgeschneiderte Lizenzierung von digitalen Technologiedaten zur Konfiguration von Laserbearbeitungsmaschinen bekannt. Das Potenzial für ihren Einsatz zur Absicherung der erworbenen Technologiedaten auf Technologiedatenmarktplätzen ist jedoch fraglos vorhanden. Dies könnte dazu beitragen, die Transparenz und Effizienz in der Laserproduktionsindustrie zu erhöhen.

Das in dieser Dissertation vorgeschlagene sichere und transparente Lizenzierungssystem zur Verwaltung von Lizenzen für erworbene Technologiedaten mithilfe von Blockchain- und Smart-Contract-Technologien kann folgende Funktionalitäten bieten:

1. Erwerben von Lizenzen: Wenn ein Lizenznehmer „Anwender“ eine Lizenz für einen digitalen Vermögenswert „Technologiedaten“ erwerben möchte, sendet er eine Anfrage an den Lizenzgeber „Technologiedatenmarktplatz“. Sobald der Kauf abgeschlossen ist, wird ein intelligenter Vertrag erstellt, der die Lizenzbedingungen abbildet. Der Vertrag wird dann mit einem eindeutigen Lizenz-Token (eine eindeutige Kennung oder ein digitaler Schlüssel) versehen und als digitaler Vertrag auf der Blockchain bereitgestellt.

2. Ausstellen von Lizenzen: Der Lizenzgeber erstellt auf Anfrage des Kunden unterschiedliche intelligente Verträge, die verschiedene Lizenzmodelle für seinen digitalen Vermögenswert abbilden. Die intelligenten Verträge enthalten Angaben zu den Lizenzbedingungen, wie zum

Beispiel Preis und Ablaufdatum. Nach der Erstellung des intelligenten Vertrags wird dieser in der Blockchain bereitgestellt.

3. Verifizieren von Lizenzen: Wenn ein Lizenznehmer die lizenzierten digitalen Inhalte nutzen möchte, wird mithilfe des intelligenten Vertrags die Identität des Lizenznehmers und die Gültigkeit der Lizenz überprüft. Nur wenn die Lizenz noch gültig ist und der Lizenznehmer die entsprechenden Zugriffsrechte hat, wird ihm der Zugriff auf die digitalen Inhalte gewährt. Das System führt dabei Protokoll über alle Aktionen und verfügt über alle zusätzlichen Informationen, die für die Nachverfolgung der Nutzung von Lizenzen erforderlich sind.

4. Erneuern von Lizenzen: Wenn die Lizenz abgelaufen ist, bietet der intelligente Vertrag dem Lizenznehmer die Möglichkeit, die Lizenz zu verlängern, indem er eine Zahlung vor dem Ablaufdatum an den Lizenzgeber tätigt. Falls die Verlängerung nicht innerhalb eines bestimmten Zeitraums erfolgt, wird der Zugriff auf die Inhalte gesperrt. Der intelligente Vertrag kann dem Lizenznehmer auch eine Benachrichtigung über das baldige Ablaufdatum der Lizenz senden, um ihn an die Verlängerung zu erinnern, bevor der Zugriff auf die Inhalte verweigert wird.

5. Übertragung von Lizenzen: Wenn ein Lizenznehmer die Lizenz auf einen anderen Nutzer übertragen möchte, kann der intelligente Vertrag die Übertragung erleichtern, indem er die eindeutige Kennung des Lizenznehmers im Vertrag aktualisiert. Dadurch wird der neue Nutzer als rechtmäßiger Inhaber der Lizenz anerkannt und er kann die damit verbundenen digitalen Inhalte nutzen.

In der vorliegenden Dissertation liegt der Fokus auf den ersten drei Funktionalitäten, während die Erneuerung und Übertragung von Lizenzen als künftige wünschenswerte Erweiterung des hier erarbeiteten Konzepts zu betrachten ist.

2.5 Modellierungsmethoden

Die Modellierungsansätze finden immer mehr Gebrauch, um die steigende Komplexität der Prozesse zu beherrschen und eine modellbasierte, verständliche und einheitliche Grundlage für die Entwicklung von Softwarelösungen zu bilden [120]. Grundsätzlich definiert *Anderl* ein Modell als eine abstrahierte Abbildung eines realen Sachverhalts [121].

Zur Datenmodellierung kommt oftmals die Methode Unified Modeling Language (UML) zum Einsatz. Die Unified Modeling Language (UML) ist ein bekannter Modellierungsstandard zur Abbildung eines System von *Objekt Management Group* (OMG) [122]. UML stellt eine objektorientierte Abbildungssprache zur Vorstellung von Systemstrukturen und –verhalten und ist sowohl für Daten- als auch für Prozessmodellierung geeignet. Hierbei werden statische Strukturen sowie dynamische Verhalten von der entwickelten System mittels mehreren Modelltypen aus verschiedenen Sichten beschrieben [123] und [124]. Die in der Realität vorkommenden Gegenstände werden in der Modellierung als Objekte bezeichnet. Gleichartige Objekte, die eine ähnliche Struktur und ein ähnliches Verhalten aufweisen, werden in einer Klasse zusammengefasst. Jede Klasse wird durch ihre Eigenschaften (Attribute, Operationen und Zusicherungen) beschrieben [125]. Die Kapselung der Struktur und des Verhaltens des Systems anhand der verschiedenen Klassen erhöht die Verständlichkeit des entwickelten Modells [126]. Dafür sollen sowohl die Datenstrukturen als auch zwischen ihnen bestehende Beziehungen definiert und in einem Informationsmodell vorgestellt.

Es gibt vier Arten von Beziehungen zwischen Klassen: Assoziation, Aggregation, Komposition und Vererbung. Assoziationen sind ungerichtete oder gerichtete, uni- oder bidirektionale Beziehungen, die mit Namen und/oder Multiplizitätsangaben bzw. Kardinalitäten beschriftet werden können. Aggregationsbeziehungen auch Hat- oder Teile-Ganzes-Beziehungen genannt, sind Beziehungsarten, bei denen Objekte als Teile einer Menge dargestellt werden können. Die Kompositionsbeziehung ist ein Sonderfall der Aggregation. Hier ist der Teil mit dem Ganzen auf existentielle Weise verknüpft. Vererbungsbeziehungen dienen der hierarchischen Darstellung, wobei Unterklassen die Eigenschaften der Oberklassen übernehmen bzw. erben. Für weitere Informationen über UML-Diagramme, -Notationen und -Applikationen sei auf die folgende Literatur hingewiesen: [92], [93], [96], [117] und [118]–[120]. Die oben vorgestellten Methoden bilden die Grundlage für die konzeptionelle Vorgehensweise in Kapitel 4.

2.6 Fazit und Potentiale

Im Unterkapitel 2.1 wurde der E-Commerce in den Blick genommen, wobei der Fokus auf den Datenmarktplätzen lag. Hierbei wurden die bekanntesten Konzepte von E-Marktplätzen sowie einige bestehende Datenmarktplätze vorgestellt. Solche Marktplätze werden in verschiedenen Branchen betrieben.

Des Weiteren wurde in Unterkapitel 2.2 das der Lasertechnologie zugrunde liegende Prinzip unter Rückgriff auf die aktuelle Literatur vorgestellt und beschrieben. Dabei wurden Applikationen vorgestellt, die in der Laserfertigung zum Einsatz kommen. Ebenfalls wurden Technologiedaten in der Laserbearbeitung analysiert, die Informationen zur Laserbearbeitungsmaschine, zum Werkstück, zum Werkzeug und zur Prozessgestaltung beinhalten. Zum besseren Verständnis der Thematik in dieser Dissertation wurde das Konzept des Technologiedatenmarktplatzes vorgestellt. Die beteiligten Teilnehmer und Systeme sowie deren Kommunikationswege und Übertragungspfade wurden ebenfalls beschrieben. Ferner wurden Schutzziele und Schutzmaßnahmen innerhalb der Vertrauensgrenzen nachgezeichnet.

Das wissensbasierte Assistenzsystem wurde dann in Unterkapitel 2.3 vorgestellt. Im vierten Unterkapitel wurden Sicherheitsmaßnahmen in der Informationstechnik dargelegt, die bei der Konzeption einer Softwareapplikation zur Unterstützung des Anwenders bei der effektiven und sicheren Nutzung von Technologiedatenmarktplätzen eine wesentliche Rolle spielen. Hierbei wurden technische Schutzmaßnahmen identifiziert, die für den vorliegenden Anwendungsfall geeignet sind. Die Beachtung der beschriebenen Sicherheitsprinzipien ist für die Funktionalität des entwickelten Konzepts essentiell. In diesem Kapitel wurden zudem allgemein die Schutzziele in der IT- Sicherheit beschrieben. Zugriffsberechtigungen und Lizenzierungen wurden ebenfalls in den Blick genommen und es wurden verschiedene Lizenzmodelle vorgestellt. Außerdem wurden verschiedene Lösungsansätze zur Lizenzierung digitaler Inhalte mittels Blockchain- und Smart-Contract-Technologien dargelegt. Letztere sind für die sichere Verwendung von Technologiedaten durch den Anwender unerlässlich. Das Kapitel schließt mit einer Darstellung der Modellierungsmethoden ab, die während der Entwicklung des Konzepts für ein Lizenzierungssystem zur Unterstützung des Anwenders bei der effektiven und sicheren Nutzung von Technologiedatenmarktplätzen verwendet werden.

Anhand einer Analyse der bestehenden Forschungsansätze wurde deutlich, dass keine Lösungen für die Unterstützung des Anwenders bei der effektiven und sicheren Nutzung von Technologiedatenmarktplätzen im Bereich der Laserbearbeitung existieren. Die Einflussfaktoren und Restriktionen bei der Nutzung der Marktplätze sowie Möglichkeiten einer effektiven Nutzung betreffend die Maschinen- und Lizenzmodellauswahl wurden bislang noch nicht untersucht. Die Sicherheit der Technologiedaten beim Anwender und vor allem die Einhaltung der Kaufbedingungen in der Nutzungsphase wurden noch nicht beleuchtet. Daher findet bislang auch keine Nutzungsverfolgung von Technologiedaten beim Anwender statt.

Was insbesondere fehlt, ist eine vollständige und systematische Konzeption zur Maschinen- und Lizenzmodellauswahl sowie zur Datensicherheit. Die Prozesse für die Verwaltung, Bereitstellung und Nutzungsverfolgung von Technologiedaten und der zugehörigen Lizenzdaten wurden noch nicht definiert. Ferner gibt es bislang noch keinen rechnerbasierten Auswahlprozess hinsichtlich der Maschinen- und Lizenzmodellauswahl für die Bearbeitung von Laserbearbeitungsaufträgen. Es bedarf eines Werkzeugs zur Erfüllung der oben genannten Funktionalitäten, auf das Anwender, die mehrere Laserbearbeitungsmaschinen von verschiedenen Herstellern besitzen, zurückgreifen können. Dieses Werkzeug wurde mit der vorliegenden Dissertation entwickelt.

3 HANDLUNGSBEDARF UND ANFORDERUNGSPROFIL

Im vorliegenden Kapitel wird zunächst der Handlungsbedarf bezüglich der Unterstützung des Anwenders bei der Nutzung von Technologiedatenmarktplätzen erörtert. Das umfasst auch die Darstellung der Bedürfnisse der Anwender hinsichtlich der sicheren Verwaltung der erworbenen Daten sowie deren rechnerbasierter Bereitstellung und ihrer Übermittlung an die Laserbearbeitungsmaschinen. Dieser Bedarf wird aus dem bereits dargestellten Stand der Technik und aus bestehenden Forschungsarbeiten hergeleitet. Im Anschluss erfolgt auf Basis des Handlungsbedarfs die Definition der mit dieser Dissertation verfolgten Ziele. Danach werden Anwendungsfälle detailliert beschrieben. Dabei werden die vorliegenden Systemgrenzen sowie die beteiligten Systemakteure in den Blick genommen. Aufbauend auf dieser Betrachtung werden mittels eines Anwendungsfalldiagramms die Anforderungen, die an das sichere Assistenzsystem gestellt werden, explizit dargestellt. Dabei wird auf die Methoden, Empfehlungen und Vorgehensweisen nach [95], [96], [135], [136], [137] und [138] zurückgegriffen. Auf dieser Grundlage wird schlussendlich die Tragfähigkeit des entwickelten Konzeptes überprüft.

3.1 Handlungsbedarf

Normalerweise werden Technologiedaten für bestimmte Werkstoffe vom Maschinenhersteller mit den Laserbearbeitungsmaschinen mitgeliefert. Für neue Werkstoffe, die sich mit dem Laser bearbeiten lassen, sind neue passende Technologiedaten erforderlich. Um an diese Daten zu gelangen, gibt es zwei Vorgehensweisen: entweder entwickelt Anwender die Technologiedaten selbst oder er bestellt diese vom Maschinenhersteller. Die Technologiedatenerstellung ist ein aufwändiger Prozess und verursacht hohe Kosten in Form von Personalaufwand sowie Material- und Maschineneinsatz. Werden die Technologiedaten vom Maschinenhersteller bezogen, herrscht wenig Flexibilität hinsichtlich des Angebots – die Daten werden zu festen Preisen verkauft, ohne Rücksichtnahme auf den Bedarf des Anwenders. Zum Beispiel macht es keinen Unterschied, ob die Daten nur einmal oder dauerhaft benötigt werden. Das treibt die Kosten nach oben. Das bedeutet, es lohnt sich, Technologiedaten über einen Technologiedatenmarktplatz zu beziehen, wo verschiedene Nutzungsmodelle angeboten werden. Hierbei eröffnen sich neue Möglichkeiten zur Reduzierung der Kosten in der Laserbearbeitung beim Anwender. Zugleich entsteht dadurch ein neues Geschäftsfeld für

Maschinenhersteller, die bereits generierte Technologiedaten auf entsprechenden Marktplätzen anbieten können. Deshalb baut jeder Maschinenhersteller seinen eigenen Technologiedatenmarktplatz auf und bietet Technologiedaten für die Bearbeitung verschiedener Werkstoffe auf seinen Laserbearbeitungsmaschinen an. Für die Vornahme der Lasereinstellungen müssen zudem Mitarbeiter mit Erfahrungswissen herangezogen werden. Aus diesen Gründen sind die richtigen Lasereinstellungen in den Technologiedaten für das Unternehmen wertvoll und müssen vor einem unbefugten Zugriff geschützt werden.

In Kapitel 2.2.4 wurde ein Konzept für den Bezug von Technologiedaten über entsprechende Marktplätze vorgestellt. Dieses innovative Konzept trägt zur Erhöhung der Wertschöpfung sowohl beim Maschinenhersteller als auch beim Anwender bei. Es bietet einen sogenannten „individuellen Datenservice“ zur Bereitstellung der Technologiedaten, dank dessen neue und innovative Geschäftsmodelle entwickelt werden können. Dieser Service dient letztlich der Verbesserung der Qualität der Leistungen, der Ausdifferenzierung des Angebots und der Reduktion der Herstellungszeiten und -kosten. Die Nutzung von Technologiedatenmarktplätzen bedarf jedoch der Einhaltung gewisser Sicherheitsstandards. Der Erwerb und die Verwendung von Technologiedaten durch den Anwender müssen in einem kontrollierten Umfeld erfolgen und die Erfüllung der Vertragsbedingungen muss sichergestellt werden. Gleichzeitig muss eine schnelle und effektive Bereitstellung der Daten erfolgen, die den Bedürfnissen der Anwender entspricht und die eine automatische Zuteilung von Technologiedaten ermöglicht.

Dem oben dargestellten Bedarf folgend ist noch kein informationsmodellbasierter Ansatz zur effektiven und sicheren Nutzung von Technologiedatenmarktplätzen vorhanden. Dies bedeutet, der Anwender muss bei Erhalt eines neuen Fertigungsauftrags für die Laserbearbeitung die passenden Technologiedaten zur Festlegung der erforderlichen Lasereinstellungen selbst über Technologiedatenmarktplätze beziehen, damit er den Fertigungsprozess starten kann. Dafür benötigt er eine Übersicht über die Verfügbarkeit der Technologiedaten. Diese enthält Informationen über den Nutzungszustand der bereits erworbenen Technologiedaten und über die zugehörigen Lizenzmodelle. Bislang kann diese Übersicht noch nicht automatisch erstellt und dem Anwender vorgelegt werden, und es besteht keine Möglichkeit, dass ihm automatisch Informationen zu den geeigneten Laserbearbeitungsmaschinen bereitgestellt werden. Demzufolge muss der Anwender zunächst die geeigneten Laserbearbeitungsmaschinen selbst bestimmen und danach die Technologiedaten für diese Laserbearbeitungsmaschinen auf den jeweiligen Technologiedatenmarktplätzen suchen und mit dem am bestem passenden Lizenzmodell bestellen. Die fehlende Unterstützung bei der sicheren und effektiven Nutzung von Technologiedatenmarktplätzen zieht aufwendige und kostenintensive Verwaltungs-,

Bereitstellungs-, Verteilungs- und Zuordnungsarbeiten nach sich. Darüber hinaus sind die Anwender aufgrund mangelnder Erfahrung mitunter mit der Entscheidung über die Maschinen- und Lizenzmodellauswahl überfordert.

Abgesehen von den bereits erwähnten Vorteilen wurde das Assistenzsystem auch in Reaktion auf das Problem entworfen, dass die auf den verschiedenen Marktplätzen erworbenen Technologiedaten durch den Anwender den Laserbearbeitungsmaschinen, Materialien, Verfahren und Nutzungsregeln zugeordnet werden müssen. Da zudem bislang keine durchgängige und übersichtliche Nachverfolgung der Nutzung von Technologiedaten möglich ist, können während des Laserbearbeitungsprozesses auftretende Fehler oder unberechtigte Nutzungen von Technologiedaten nicht nachvollzogen werden. Weiterhin ist es nicht möglich, die passenden Lizenzmodelle für die angebotenen Technologiedaten automatisch zu ermitteln. Zurzeit kann diese Auswahl nur fallbezogen und unter Rückgriff auf erfahrungsbasierte Einschätzungen erfolgen. Es gibt hierfür keinen Algorithmus, was die verfahrensindividuelle Wahl zwischen mehreren Laserbearbeitungsmaschinen oder Lizenzmodellen zu einem mühseligen Unterfangen macht. Das im Rahmen dieser Dissertation entwickelte Assistenzsystem liefert dem Anwender bezüglich der oben genannten Probleme wertvolle Handlungsempfehlungen und spart damit Kosten und Zeit.

3.2 Zieldefinition

Ziele müssen verständlich und eindeutig formuliert werden, damit zweifelsfrei feststellbar ist, ob sie erreicht wurden [129].

Das Ziel der vorliegenden Dissertation ist die Entwicklung und Umsetzung eines Assistenzsystems zur Unterstützung des Anwenders bei der sicheren und effektiven Nutzung von Technologiedatenmarktplätzen im Feld der Lasermarkierung.

Das konzipierte System stellt dem Anwender Handlungsempfehlungen zur Maschinen- und Lizenzmodellauswahl zur Verfügung. Die Datensicherheit – vor allem der Schutz von Technologiedaten – steht hierbei im Fokus und soll bei der Konzeption durchgehend in Betracht gezogen werden. Letztlich soll eine effektive, sichere und vertrauenswürdige Konzeption vorliegen, die den Anwender bei der Nutzung von Technologiedatenmarktplätzen unterstützt und damit die Produktivität in der Lasermarkierung erhöht.

Das erste Teilziel dieser Dissertation besteht in der Bereitstellung von Handlungsempfehlungen zur Maschinenauswahl im Rahmen der Abarbeitung eines Auftrages. Das zweite Teilziel lautet, den Anwender bei der Auswahl passender Lizenzmodelle zu unterstützen, sodass der Erwerb von Technologiedaten effektiv und schnell vonstattengehen kann. Das dritte Teilziel ist die Gewährleistung einer sicheren Übermittlung und Nutzung der Daten.

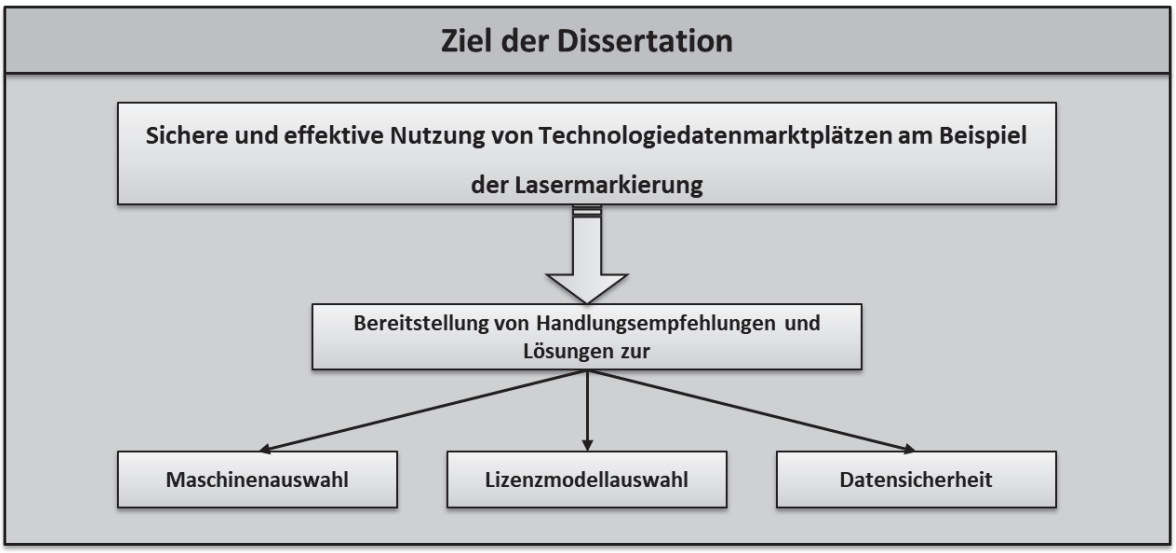

Abbildung 3-1: Mit dieser Dissertation verfolgte Ziele (eigene Darstellung)

Zur Erreichung der oben dargestellten Ziele werden alle Arbeitsschritte im Umgang mit den erworbenen Technologiedaten beschrieben. Weiterhin werden alle im Zuge dessen eingehenden oder ausgehenden Informationen identifiziert. Die erworbenen Technologiedaten und die zugehörigen Lizenzen werden hierbei klar den verschiedenen Maschinen und den Fertigungsprozessen zugeordnet. Dazu werden die Technologiedaten mit deren Nutzungsbedingungen verknüpft. Über diese Verknüpfung werden die Informationen bezüglich der Nutzungszustände von Technologiedaten vor und nach dem Fertigungsprozess bereitgestellt und gespeichert. Aufbauend auf der entwickelten Konzeption wird ein rechnergestütztes Assistenzsystem entwickelt, das dem Anwender die durchgängige und sichere Verwaltung von Technologiedaten und der zugehörigen Lizenzen ermöglicht. Die effektive Bereitstellung von Technologiedaten bedeutet eine erhebliche Erleichterung des auftragsspezifischen Entscheidungsprozesses hinsichtlich der Maschinen- und Lizenzmodellauswahl. Außerdem kann so die Technologiedatennutzung nachvollzogen werden; die für die erworbenen Daten geltenden Nutzungsregeln können laufend überprüft, kontrolliert und aktualisiert werden. Im Rahmen der vorliegenden Dissertation wird also eine Konzeption zur Unterstützung des Anwenders bei der effektiven, sicheren und vertrauenswürdigen Nutzung

von Technologiedatenmarktplätzen entworfen. Zudem werden Werkzeuge zur effektiven Bereitstellung von Technologiedaten sowie zur Kontrolle und Nachverfolgung von deren Nutzung bereitgestellt. In den nächsten Unterkapiteln werden Anwendungsfälle beschrieben. Auf dieser Grundlage wird das Profil der Anforderungen, die an das Assistenzsystem gestellt werden, erarbeitet.

3.3 Betrachtete Anwendungsfälle

Zur vollständigen Spezifikation des in dieser Dissertation behandelten Problems werden nun einige Anwendungsfälle für das entwickelte System dargelegt. Hierfür werden Anwendungsfalldiagramme (engl. Use Case Diagrams) mittels der Unified Modeling Language (UML) entworfen. Die geleisteten Dienste werden dabei innerhalb eines definierten Kontextes eines Systems modelliert [128]. Anwendungsfälle (engl. Use Cases) beschreiben ein System statisch als auch dynamisch. Hierfür werden die Interaktionen der Akteure (auch „Stakeholder“ genannt) mit dem System identifiziert [139]. Ein Anwendungsfall wird in der Norm ISO/IEC 19505-2:2012 folgendermaßen definiert:

> ***A use case is the specification of a set of actions performed by a system, which yields an observable result that is, typically, of value for one or more actors or other stakeholders of the system [140].***

Das Systemverhalten, die Systemfunktionalitäten und die Grenzen des Systems werden aus Sicht der beteiligten Akteure grafisch modelliert, ohne dass jedoch auf Details zur Implementierung (wie zum Beispiel Datenstrukturen und Algorithmen) eingegangen wird. Das bedeutet eine Hilfestellung beim Definieren der Systemanforderungen [131]. Das Anwendungsfalldiagramm wird im Bereich der Softwareentwicklung zur Problemspezifikation angewandt [130]. Hierdurch werden Akteure, Anwendungsfälle und Beziehungen zwischen diesen Elementen innerhalb der festgelegten Systemgrenzen beschrieben und damit das Systemverhalten deutlicher dargestellt. Es dient dazu, Akteure und Anwendungsfälle sowie die Beziehungen zwischen diesen Elementen innerhalb definierter Systemgrenzen zu beschrieben und damit das Systemverhalten deutlicher darzustellen. Das Anwendungsfalldiagramm, das in der Unified Modeling Language (UML) verfasst wurde, wird in Abbildung 3-2 dargestellt.

In den folgenden Abschnitten werden Anwendungsfälle sowie dazugehörige Systemgrenzen, Systemakteure und Beziehungen definiert.

3.3.1 Systemgrenze

Die Systemgrenze wird durch den Prozess der Laserfertigung abgesteckt. Dieser umfasst auch die durchgängige Informationsverarbeitung, beginnend mit dem Starten der Bearbeitung eines neuen Auftrags bis hin zur Fertigung des Werkstücks auf der Laserbearbeitungsmaschine. Die Handlungen der beteiligten Akteure innerhalb der Systemgrenzen werden in Form von Anwendungsfällen dargestellt. Über die Systemgrenzen hinweg werden Informationen transferiert; es findet ein Austausch mit verschiedenen Akteuren, beispielsweise mit den Betreibern von Technologiedatenmarktplätzen, statt.

3.3.2 Systemakteure

Die Akteure werden in diesem Zusammenhang auch als „Systemnutzer“ bezeichnet. Diese können eindeutig identifiziert werden, und es kann sich dabei um Personen oder um andere Systeme handeln. Vier Akteure werden im Kontext dieser Dissertation in den Blick genommen: „Assistenzsystem“, „Anwender“, „Technologiedatenmarktplatz“ und „Laserbearbeitungsmaschine“.

Das „Assistenzsystem“ ist rechnergestützt und führt die gewünschten Funktionalitäten aus, indem es eingegebene Informationen verarbeitet, um dem Anwender die benötigten Informationen bereitzustellen. Diese Anwendung kommuniziert über definierte Schnittstellen mit den bestehenden Systemen in der Fertigungsumgebung.

Der „Anwender“ repräsentiert im Anwendungsfalldiagramm den Maschinenbetreiber, der mehrere Laserbearbeitungsmaschinen von verschiedenen Herstellern besitzt und Technologiedaten für verschiedene Fertigungsaufträge auf den Technologiedatenmarktplätzen bedarfsgerecht erwirbt. Er initiiert die rechnergestützte Vorbereitung des Laserfertigungsprozesses in der Planungsphase unter Berücksichtigung der Besonderheiten des Auftrags. Für die Erledigung des Auftrags benötigt er Technologiedaten, über die er entweder bereits verfügt oder die er noch beziehen muss.

Der „Technologiedatenmarktplatz“ ist ein vertrauenswürdiges System, das korrekte Marktvorgänge und sichere Geschäftsprozesse gewährleistet. Auf dem Marktplatz gelten je nach gewähltem Kaufmodell verschiedene Nutzungsregeln, die den erworbenen Technologiedaten sowie dem Anwender, in diesem Fall dem Marktplatzkunden, zugeordnet werden. Der Anwender kann als Kunde auf mehreren Technologiedatenmarktplätzen gleichzeitig operieren.

Nach einem erfolgreich abgeschlossenen Kaufvorgang werden die Technologiedaten dem Anwender zusammen mit den vereinbarten Nutzungsregeln bzw. der generierten Lizenz zur Verfügung gestellt.

Die „Laserbearbeitungsmaschine“ ist eine von mehreren beim Anwender vorhandenen Geräten, die von unterschiedlichen Herstellern stammen. Diese Maschinen basieren auf verschiedenen Lasertechnologien und haben entsprechend unterschiedliche Eigenschaften. Auf diesen Laserbearbeitungsmaschinen können diverse Fertigungsprozesse vonstattengehen. Dafür werden Technologiedaten zur Einstellung der Maschine zu Beginn des Fertigungsprozesses benötigt.

Auf der Abbildung 3-2 ist das Anwendungsfalldiagramm dargestellt, wobei die Akteure außerhalb der Systemgrenzen symbolisch abgebildet und mit den Anwendungsfällen verbunden sind.

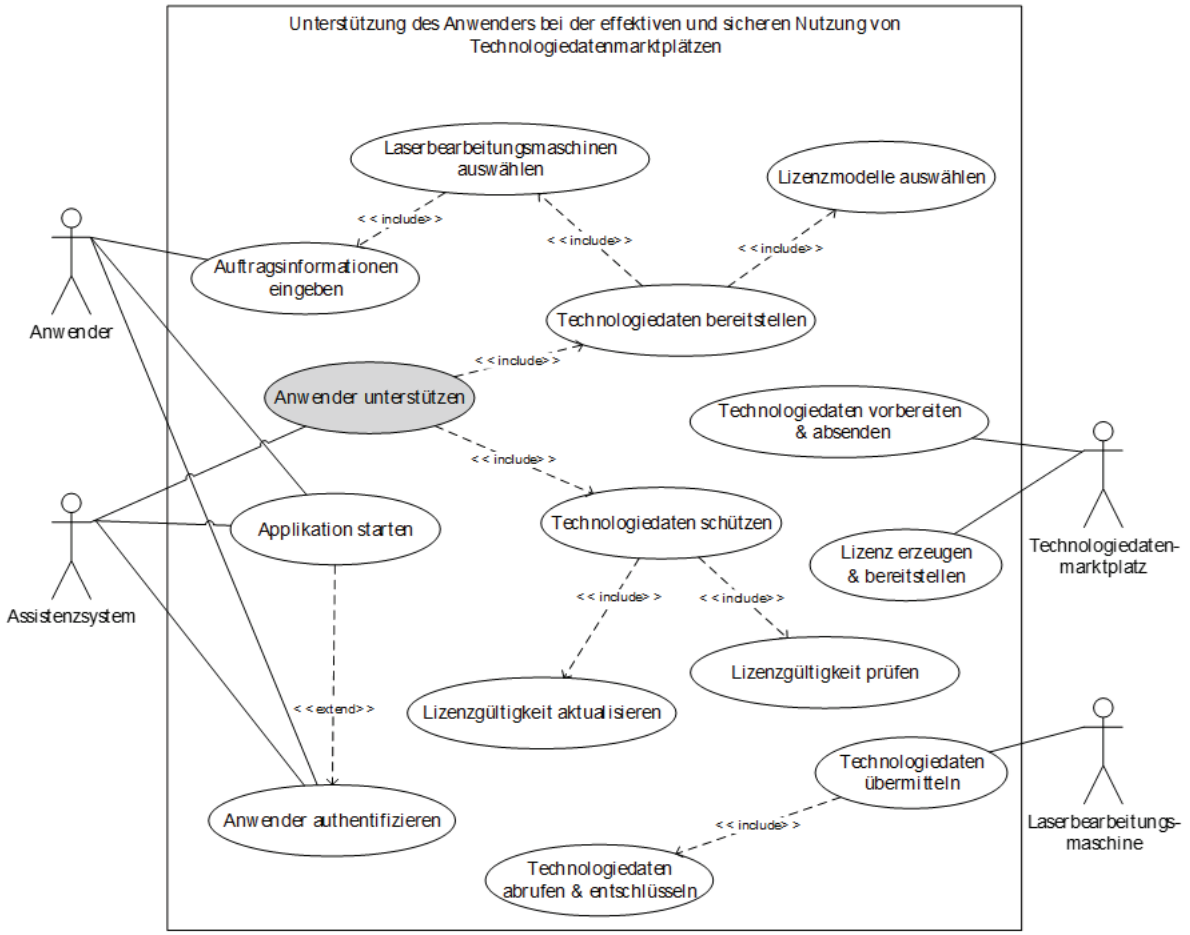

Abbildung 3-2: Anwendungsfalldiagramm zur Unterstützung des Anwenders bei der effektiven und sicheren Nutzung von Technologiedatenmarktplätzen (eigene Darstellung)

Nun wird das Systemverhalten anhand der definierten Anwendungsfälle vorgestellt.

3.3.3 Systemverhalten

Das Systemverhalten wird durch Anwendungsfälle beschrieben. Anwendungsfälle sind Aktionen, die das System für einen oder mehrere Akteure vorsieht [141]. Das Anwendungsfalldiagramm, siehe Abbildung 3-2, besteht aus vierzehn Anwendungsfällen, welche zum Teil mit anderen Anwendungsfällen in Beziehung stehen. Einige Anwendungsfälle weisen untereinander Include- oder Extend-Beziehungen auf.

Bei einer Include-Beziehung (auf Deutsch: Enthält-Beziehung) ruft ein Anwendungsfall immer einen weiteren Anwendungsfall auf, der dann mit durchgeführt wird. Bei einer Extend-Beziehung (auf Deutsch: Erweiterungsbeziehung) kann ein Anwendungsfall optional durch einen anderen Anwendungsfall erweitert werden [121] und [129]. Diese Anwendungsfälle wurden bereits in der Abbildung 3-2 dargestellt und werden im Folgenden detailliert beschrieben. Als Hauptanwendungsfall wurde „Anwender unterstützen“ identifiziert. Dieser Fall gliedert sich in die zwei Anwendungsfälle „Technologiedaten bereitstellen“ und „Technologiedaten schützen“.

- **Anwender unterstützen:** Der Hauptanwendungsfall ist mit den zwei Anwendungsfällen „Technologiedaten bereitstellen“ und „Technologiedaten schützen“ über eine Include-Beziehungen verknüpft. Diese beiden Anwendungsfälle stehen wiederum mit anderen Anwendungsfällen in Verbindung. Alle in den verschiedenen Prozessschritten anfallenden Informationen werden von der Applikation im Anwendungsfall „Anwender unterstützen“ verarbeitet. Die Technologiedaten und die zugehörigen Lizenzen werden hierbei gespeichert, aktualisiert und bei berechtigten Anfragen bereitgestellt.

- **Technologiedaten bereitstellen:** In diesem Anwendungsfall werden benötigte Technologiedaten für die Bearbeitung eines neuen Antrags zur Verfügung gestellt. Wenn diese Technologiedaten nicht vorhanden sind, müssen sie auf dem entsprechenden Marktplatz mittels eines passenden Lizenzmodells erworben werden. Dieser Anwendungsfall enthält die zwei Anwendungsfälle „Laserbearbeitungsmaschine auswählen“ und „Lizenzmodelle auswählen“.

- **Laserbearbeitungsmaschinen auswählen:** Dieser Anwendungsfall steht über eine Include-Beziehung mit dem Anwendungsfall „Technologiedaten bereitstellen“ in Verbindung. Der Anwender gibt die Informationen aus dem Fertigungsauftrag ein, beispielsweise Angaben zum Fertigungsprozess und zu den Materialeigenschaften. Deshalb steht dieser Anwendungsfall über eine Include-Beziehung mit dem Anwendungsfall „Auftragsinformationen eingeben“ in Verbindung. Das System unterstützt den Anwender in der Folge bei der Auswahl der geeigneten Laserbearbeitungsmaschinen für die Bearbeitung eines bestimmten Auftrags.

- **Lizenzmodelle auswählen:** Eine Include-Beziehung verbindet diesen Anwendungsfall mit dem Anwendungsfall „Technologiedaten bereitstellen“. Hierbei werden dem Anwender Handlungsempfehlungen bezüglich der am besten passenden Lizenzmodelle gegeben. Es wird demnach auf die Effektivität der gewählten Lizenzmodelle geachtet, die sich anhand mehrerer Kriterien, zum Beispiel anhand der Auftragsinformationen und der Anwendererfahrungen, bestimmen lässt.

- **Auftragsinformationen eingeben:** Dieser Anwendungsfall steht über eine Extend-Beziehung mit dem Anwendungsfall „Laserbearbeitungsmaschinen auswählen“ in Verbindung. In diesem Anwendungsfall startet der Anwender die Abwicklungsvorbereitung eines Auftrags (z. B. Laserschneiden vom Titanblech). Wie das Starten vonstattengeht, ist abhängig vom Laserbearbeitungsverfahren, das vom Anwender gewählt wurde. Im Auftrag sind die Anforderungen an das Fertigungsverfahren bezüglich Material, Geometrie und Lieferungsumfang festgehalten. Aufbauend auf diesen Anforderungen werden spezielle Technologiedaten für die Durchführung des Fertigungsprozesses benötigt.

- **Technologiedaten schützen:** Dieser auslösende Anwendungsfall umfasst die sichere Nutzung von erworbenen Technologiedaten unter den vereinbarten Lizenzvereinbarungen. Diese Technologiedaten und die zugehörigen Lizenzen werden in einer strukturierten Form gespeichert. Die Daten und die Verbindungen werden einer Laserbearbeitungsmaschine zugeordnet und die dazugehörigen Nutzungsregeln sowie die Nutzungszustände werden übersichtlich dargestellt. Dabei sollen der Schutz der Technologiedaten und die Einhaltung der festgelegten Nutzungsregeln zur Vergabe von Daten sichergestellt werden. Dieser Anwendungsfall enthält die zwei Anwendungsfälle „Lizenzgültigkeit prüfen“ und „Lizenzgültigkeit aktualisieren“.

- **Lizenzgültigkeit prüfen:** Dieser Anwendungsfall wird schlagend, wenn die benötigten Technologiedaten dem Anwender bereits vorliegen. Im Rahmen dieses Falls wird die Gültigkeit der Lizenz und der darin enthaltenen vereinbarten Nutzungsregeln geprüft. Sind die Nutzungsregeln noch gültig, werden Technologiedaten für die Laserbearbeitungsmaschine bereitgestellt. Ist das nicht der Fall, muss eine neue Lizenz erworben werden.

- **Lizenzgültigkeit aktualisieren:** Eine Lizenz beinhaltet die vereinbarten Nutzungsregeln und dient zu deren Durchsetzung beim Anwender. Oder anders ausgedrückt: Sie stellt eine kontrollierte Nutzung der Technologiedaten sicher. Das ist wesentlich für die Zuverlässigkeit und Vertrauenswürdigkeit des Gesamtkonzepts. Die Nutzungsregeln müssen stets nach jeder Anwendung aktualisiert werden.

- **Technologiedaten abrufen & entschlüsseln:** Nach einem erfolgreichen Erwerb liegen Technologiedaten in einer verschlüsselten Form auf einem freigegebenen Ordner beim Anwender. Um den Laserbearbeitungsprozess starten zu können, müssen Technologiedaten in einer für die Maschine lesbaren Weise zur Verfügung gestellt werden. Das beinhaltet den Abruf der verschlüsselten Technologiedaten und des Schlüssels, der dazu dient, sie zu decodieren. Dieser Anwendungsfall steht über eine Include-Beziehung mit dem Anwendungsfall „Technologiedaten übermitteln“ in Verbindung.

- **Technologiedaten übermitteln:** Das entwickelte System soll in der Lage sein, Technologiedaten an die jeweilige Laserbearbeitungsmaschine zu senden, um das Starten der Laserbearbeitung auf der gewählten Laserbearbeitungsmaschine zu ermöglichen.

- **Applikation starten:** In diesem Anwendungsfall wird die Software vom Anwender in Betrieb genommen. Zunächst werden ihm die verfügbaren Funktionen angezeigt. Dieser Anwendungsfall weist eine Extend-Beziehung mit den Anwendungsfall „Anwender authentifizieren“ auf.

- **Anwender authentifizieren:** Bei diesem Anwendungsfall legitimiert sich der Anwender durch Eingabe seiner Login-Daten. Es folgt der Authentifizierungsprozess mithilfe des Anmeldeprozesses über die Applikation. Hierbei werden die eingegebenen Daten auf Korrektheit geprüft, um die Identität des Anwenders überprüfen zu können. Dieser

Anwendungsfall ist für die Erfüllung der Sicherheitsanforderungen entscheidend, denn hierdurch können Datenmanipulation und unerwünschte Vorfälle oder Zugriffe weitgehend vermieden werden.

- **Lizenz erzeugen & bereitstellen:** In diesem Anwendungsfall wird eine Lizenz für die erworbenen Technologiedaten unter Berücksichtigung der vereinbarten Nutzungsregeln erzeugt. Diese Lizenz wird dann dem Anwender zur Verfügung gestellt. Der Anwender kann eine seiner Lizenzen bei Bedarf abrufen, zum Beispiel über eine bereitgestellte Schnittstelle.

- **Technologiedaten vorbereiten & absenden:** Nach der erfolgreichen Abwicklung des Kaufs der Technologiedaten auf dem Marktplatz werden sie zunächst vorbereitet und dem Anwender in einer sicheren Form übermittelt. Dies umfasst auch die Verschlüsselung und Signierung der Technologiedaten sowie das Senden zum Anwender über einen sicheren Kanal.

Nach dem die Akteure, die Grenzen und das Verhalten des Systems vorgestellt wurden, werden im nachfolgenden Kapitel die die Anforderungen an die Entwicklung des Assistenzsystems zur Unterstützung des Anwenders bei der effektiven und sicheren Nutzung von Technologiedatenmarktplätzen festgelegt.

3.4 Anforderungsprofil

Aufbauend auf dem analysierten Stand der Technik, dem hergeleiteten Handlungsbedarf und der Zieldefinition sowie auf der Beschreibung der Anwendungsfälle wird nachfolgend das Anforderungsprofil zur Entwicklung einer Konzeption für die Unterstützung des Anwenders bei der effektiven und sicheren Nutzung von Technologiedatenmarktplätzen erstellt. Das Anforderungsprofil gliedert sich in die drei Anforderungsbereiche „Anforderungen an die Methode zur Konzeptionsentwicklung“, „Anforderungen an das Informationsmodell“ und „Anforderungen an die Implementierung“.

3.4.1 Anforderungen an die Methode

Für die Entwicklung eines Konzepts zur Unterstützung des Anwenders bei der Nutzung von Technologiedatenmarktplätzen bedarf es einer passenden Methode. Diese Methode soll eine

Vorgehensweise hervorbringen, auf die der Anwender zurückgreifen kann. Dies soll ihm den sicheren Zugang zu Technologiedaten erleichtern. Zudem soll ihm auf diesem Weg eine rechnerunterstützte Auswahlhilfe bei der Maschinen- und Lizenzmodellauswahl zur Verfügung gestellt werden. Und nicht zuletzt sollen ihm wirksame Sicherheitsmaßnahmen bezüglich der Verteilung und Nutzung der Technologiedaten vorgeschlagen werden.

Anforderung 1:

> *Die Methode muss eine Vorgehensweise definieren, anhand der ein auftragsspezifischer Ansatz zur Auswahl der geeigneten Laserbearbeitungsmaschinen entwickelt werden kann.*

Die Methode muss für jeden Auftrag die Auswahl geeigneter Laserbearbeitungsmaschinen auf Basis der Fertigungsprozess- und Werkstückinformationen ermöglichen. Nachdem der Anwender diese Informationen eingegeben hat, sollen sie verarbeitet werden und es soll eine Liste der geeigneten Laserbearbeitungsmaschinen bereitgestellt werden.

Anforderung 2:

> *Die Methode muss eine Vorgehensweise definieren, anhand der eine rechnerbasierte Maschinenauswahl ermöglicht wird.*

Die Unterstützung des Anwenders hinsichtlich der Auswahl einer Maschine zur Bearbeitung eines Auftrags ist von großer Bedeutung. Die hier gewählte Methode soll das leisten. Das geschieht rechnerbasiert und automatisch auf Basis der im Fertigungsauftrag enthaltenen Informationen.

Anforderung 3:

> *Die Methode muss eine Vorgehensweise definieren, anhand der ein effektiver und rechnerbasierter Ansatz zur Auswahl der passenden Lizenzmodelle realisiert werden kann.*

Auf den Technologiedatenmarktplätzen werden verschiedene Lizenzmodelle angeboten. Im Rahmen des Erwerbs der benötigten Daten muss entschieden werden welches Lizenzmodell für den Kauf am besten zum jeweiligen Auftrag passt. Die Methode zur Unterstützung des Anwenders bei der Bereitstellung von Technologiedaten muss effektive Handlungsempfehlungen hinsichtlich der Auswahl eines Lizenzmodells anbieten. Dafür kann

auf einen rechnerbasierten Ansatz zurückgegriffen werden, wobei die benötigten Informationen mittels eines vordefinierten Mechanismus verarbeitet werden.

Anforderung 4:

> *Die Methode muss eine Vorgehensweise definieren, anhand der eine gezielte Zuteilung der Daten zu den Laserbearbeitungsmaschinen möglich ist.*

In der Konzeption wird davon ausgegangen, dass der Anwender unterschiedliche Typen von Laserbearbeitungsmaschinen von verschiedenen Herstellern verwendet. Je nach Auftrag und Laserbearbeitungsmaschine werden unterschiedliche Technologiedaten benötigt. Die Zuteilung diese Technologiedaten zur jeweiligen Maschine muss bedarfsgerecht und kontrolliert sowie unter Einhaltung von Sicherheitsstandards vonstattengehen.

Im folgenden Unterkapitel werden die Anforderungen an die Methode bezüglich der Datensicherheit untersucht und festgelegt. Dabei müssen die Sicherheitsanforderungen vollständig, präzise, angemessen und widerspruchsfrei bezüglich anderer Anforderungen formuliert werden [142], [143].

3.4.2 Sicherheitsanalyse

Bevor die Sicherheitsanforderungen definiert werden, wird eine Sicherheitsanalyse durchgeführt. Dabei werden die Sicherheitsziele sowie mögliche Bedrohungen identifiziert.

Die Technologiedaten sollen vom Anwender gemäß den vereinbarten, in Form von Lizenzen festgeschriebenen Nutzungsregeln verwendet werden. Zu Beginn werden die Schutzbedürfnisse anhand der Schutzziele identifiziert. Im Kontext der in dieser Dissertation festgelegten Systemgrenzen sind die Technologiedaten und die dazugehörigen Lizenzen für die entwickelte Konzeption von großer Bedeutung. Daher müssen sie durch geeignete Sicherheitsmaßnahmen geschützt werden.

Die Technologiedaten müssen also vertraulich behandelt werden, sie müssen vor unbefugten Zugriffen abgeschirmt werden. Außerdem soll die Integrität der Technologie- und Lizenzdaten sichergestellt werden. Essentiell ist hierbei die Korrektheit der Lizenzdaten, sprich der vereinbarten Nutzungsregeln, die vom Anwender nicht missbraucht oder manipuliert werden dürfen. Das bedeutet, dass die Schutzziele Vertraulichkeit und Integrität von großer Bedeutung

bezüglich der Datensicherheit sind. Das Schutzziel Authentizität hat ebenfalls eine bedeutsame Rolle: die Identität des Anwenders muss nachgewiesen werden. Das Schutzziel Autorisierung ist gleichermaßen bedeutsam, wobei die Möglichkeit des Zugriffs auf Daten durch Berechtigungsregeln eingeschränkt wird. Das Schutzziel Verbindlichkeit bzw. Nicht- Abstreitbarkeit spielt auch eine wesentliche Rolle. Hierbei soll nach dem erfolgreichen Erwerb ein digitaler Vertrag zwischen dem Marktplatz und dem Anwender abgeschlossen werden. Damit wird gegenüber Dritten eindeutig nachgewiesen, dass die beiden Parteien Lizenzgeber „Technologiedatenmarktplatz“ und Lizenznehmer „Anwender“ an einer entsprechenden Geschäftsbeziehung teilhaben. Die Technologiedaten sowie die Lizenzdaten sollen beim Bedarf zur Verfügung gestellt werden, um die Durchführung des Fertigungsprozesses zu ermöglichen. Daher spielt das Schutzziel Verfügbarkeit insbesondere für die Laserbearbeitungsprozesse eine große Rolle. Die oben diskutierten Schutzziele werden in den kommenden Schritten bei der Bedrohungsmodellierung berücksichtigt.

Die Methode *STRIDE* wurde für die Bedrohungsmodellierung sowohl für Cyber- als auch für cyber-physikalischen Systeme erfolgreich angewandt – siehe dazu beispielsweise [144], [145] und [146]. In der Bedrohungsanalyse werden die Bedrohungen identifiziert. Hierfür dient der Gefährdungskatalog des *BSI* (Bundesamt für Sicherheit in der Informationstechnik) [147] unter Berücksichtigung der Systemgrenzen. Untenstehend findet sich eine Liste der darin beschriebenen Bedrohungen und es wird deren Bedeutung für die Konzeption dieser Dissertation dargelegt:

- Ausspähen von Informationen: Technologiedaten und deren Lizenzen könnten durch verschiedene Angreifer abgefangen werden, die sich zum Beispiel Wettbewerbsvorteile verschaffen wollen. Ziel ist eine unberechtigte Nutzung oder auch der Verkauf dieser Technologiedaten.

- Abhören: Technologiedaten könnten im Zuge der unverschlüsselten Datenübertragung gezielt erbeutet werden. Deshalb ist es wichtig, diese Daten in verschlüsselter Form zu übertragen.

- Offenlegung schützenswerter Informationen: Die Technologiedaten dürfen vom Anwender nicht an Dritte weitergegeben bzw. weiterverkauft werden. Der Verlust dieser Daten kann in Verbindung mit den zugehörigen Lizenzdaten negative Folgen für den Anwender haben: Das Vertrauensverhältnis zum Technologiedatenanbieter kann gestört

werden und es können dem Anwender daraus rechtliche und/oder finanzielle Probleme erwachsen.

- Manipulation von Informationen: Eine unberechtigte Veränderung von erworbenen Technologiedaten kann zu Schäden sowohl an den Laserbearbeitungsmaschinen als auch an den bearbeiteten Bauteilen führen. Die Manipulation von Lizenzbedingungen ermöglicht zudem eine unberechtigte Nutzung von Technologiedaten und kann zu rechtlichen Auseinandersetzungen mit dem Anbieter führen. Solche Manipulationen an Technologiedaten sowie an Lizenzen müssen daher unterbunden werden.

- Verstoß gegen Regelungen: Ist das Schutzkonzept bezüglich der Unterstützung des Anwenders bei der sicheren Nutzung von Technologiedatenmarktplätzen nicht ausgereift, kann es zu Verstößen gegen mit Technologiedatenmarktplätzen getroffenen vertraglichen Vereinbarungen, zu einer unberechtigten Nutzung der Technologiedaten kommen.

- Identitätsdiebstahl: Werden Personendaten zur Identifizierung der Nutzer gestohlen, können die Angreifer einen berechtigten Zugriff auf die Technologiedaten vortäuschen und so alle dem Anwender zugeordneten Funktionen und Rechte für sich in Anspruch nehmen.

- Ausfall oder Störung von Kommunikationsnetzen: Eine Unterbrechung der Kommunikationswege in der Netzwerkverbindung des Anwenders kann zum Abbruch bzw. zu einer Verzögerung des Produktionsprozesses führen, was mit hohen Verlusten verbunden ist.

Nachdem die Schutzziele und Bedrohungen innerhalb der Systemgrenzen untersucht wurden, werden nun die Sicherheitsanforderungen an die Methode festgelegt. Unter einer Sicherheitsanforderung wird Folgendes verstanden:

> ***Als Sicherheitsanforderung werden Anforderungen für den organisatorischen, personellen, infrastrukturellen und technischen Bereich bezeichnet, deren Erfüllung zur Erhöhung der Informationssicherheit notwendig ist bzw. dazu beiträgt [148].***

Eine Sicherheitsanforderung fasst letztendlich die für das festgelegte Niveau an Informationssicherheit erforderlichen Handlungen und Maßnahmen zusammen. Die Sicherheit

der Technologiedaten ist eine Voraussetzung für einen individuellen und kostengünstigen Bezug der Daten auf den Technologiedatenmarktplätzen. Die Methode definiert die Vorgehensweise zur Gewährleistung der Sicherheit der Technologiedaten und Lizenzen. Hierbei sollen die Schutzziele Vertraulichkeit, Integrität, Authentizität, Autorisierung, Verbindlichkeit und Verfügbarkeit als Grundlage für die Konzeption beachtet werden. Nun werden die Sicherheitsanforderungen an die Methode definiert und vorgestellt:

Anforderung 5:

> *Die Methode muss eine Vorgehensweise definieren, anhand der die Vertraulichkeit der verarbeitenden Daten gewährleistet werden kann.*

Anforderung 6:

> *Die Methode muss eine Vorgehensweise definieren, anhand der die Integrität der Daten sichergestellt wird.*

Anforderung 7:

> *Die Methode muss eine Vorgehensweise definieren, welche die Authentifizierung der Kommunikationspartner ermöglicht.*

Technologiedaten sind wertvoll, da deren Generierung aufwändig ist und viel Erfahrungswissen erfordert. Deshalb muss die Sicherheit der Technologiedaten während der Nutzung an den Laserbearbeitungsmaschinen innerhalb der Systemgrenzen gewährleistet werden. Es bedarf also einer sicheren Umgebung für die Speicherung und Verarbeitung dieser Daten. Außerdem müssen die Daten ungefährdet zwischen den Systemen übertragen werden können. Hierfür sorgen etablierte Sicherheitsprotokolle und -methoden. Die erworbenen Technologiedaten müssen vertraulich und integer behandelt und verarbeitet werden. Außerdem sollen Datenmanipulation und unberechtigte Änderungen ausgeschlossen werden. Ferner müssen ausreichende Möglichkeiten der Authentifizierung bestehen. Zusammengefasst soll die Methode die Gestaltung einer sicheren Umgebung hinsichtlich der Übertragung und Nutzung von Technologiedaten und der zugehörigen Lizenzen ermöglichen.

Anforderung 8:

Die Methode muss eine Vorgehensweise definieren, anhand der eine Überprüfung der Nutzungsregeln vor dem Laserbearbeitungsprozess sichergestellt wird.

Anforderung 9:

Die Methode muss eine Vorgehensweise definieren, die die Nachverfolgung von Änderungen während der Datenverarbeitung ermöglicht.

Für eine kontrollierte Nutzung der Technologiedaten muss die Methode einen Ansatz definieren, der die unberechtigte Nutzung dieser Daten unterbindet sowie die Nachverfolgung der Technologiedatennutzung ermöglicht. Hierbei werden alle Datenverarbeitungsprozesse registriert, um Änderungen von Seiten des Anwenders nachzuverfolgen. Ferner soll der Stand der Nutzungsregeln vor und nach der Nutzung überprüft werden.

Anforderung 10:

Die Methode muss eine Vorgehensweise definieren, anhand der die Verfügbarkeit von Daten sichergestellt wird.

Es ist von immanenter Bedeutung, dass die Daten zur rechten Zeit verfügbar sind. Nur so kann die Produktion aufrechterhalten werden, womit Verluste vermieden werden können. Um die Produktion effizient zu gestalten, muss zeitsparend gearbeitet werden. Fehlen Technologiedaten, gerät der Fertigungsprozess ins Stocken. Die Methode soll eine Vorgehensweise ermitteln, dank der die Technologiedaten bei jeder Anfrage unmittelbar zur Verfügung gestellt werden.

Anforderung 11:

Die Methode muss eine Vorgehensweise definieren, anhand der eine Überprüfung der Verbindlichkeit bzw. Nicht- Abstreitbarkeit sichergestellt wird.

Nach einem erfolgreichen Erwerb von Technologiedaten auf dem Marktplatz soll ein digitaler Vertrag zwischen dem Marktplatz und dem Anwender abgeschlossen werden. Das ist essentiell für diese Konzeption, um gegenüber Dritten eindeutig nachzuweisen, dass die Lizenzgeber und Lizenznehmer an einer entsprechenden Geschäftsbeziehung teilhaben.

Anforderung 12:

Die Methode soll eine Vorgehensweise definieren, anhand der eine Autorisierungsstrategie zur Verwaltung der Zugriffsrechte von Nutzern erarbeitet wird.

Technologiedaten sollen beim Anwender sicher abgespeichert und verarbeitet werden. Unberechtigte Zugriffe auf die Technologiedaten müssen ausgeschlossen werden. Das Lesen, Ändern und Verwenden der Daten muss autorisierten Nutzern vorbehalten sein. Dafür soll eine Berechtigungsstrategie beim Anwender ausgearbeitet werden, welche die Nutzerrollen und -rechte definiert.

3.4.3 Anforderungen an das Informationsmodell

Die Unterstützung des Anwenders bei der Nutzung von Technologiedatenmarktplätzen für die Laserbearbeitung funktioniert mittels einer Softwareapplikation. Diese Applikation verarbeitet Informationen mit dem Ziel einer auftragsspezifischen Bereitstellung von Technologiedaten. Ferner werden Handlungsempfehlungen zur Maschinen- und Lizenzmodellauswahl vorbereitet und zur Verfügung gestellt. Die formalen Spezifikationen der erforderlichen Datenstrukturen und von deren Beziehungen und Abhängigkeitsverhältnissen werden durch ein Informationsmodell in klarer und strukturierter Weise repräsentiert. Weiterhin werden die Informationen zur Datensicherheit und zu den zugehörigen Nutzungsregeln strukturiert beschrieben und dargestellt. Die Anforderungen, die an die digitale Repräsentation des Informationsmodells gestellt werden, werden nachfolgend dargelegt:

Anforderung 13:

Das Informationsmodell muss maschinen-, material-, auftrags- und prozessrelevante Informationen enthalten.

Anforderung 14:

Das Informationsmodell muss die Datenstrukturen und -beziehungen formal abbilden.

Anforderung 15:

Das Informationsmodell muss die Verknüpfungen zwischen den Datenstrukturen übersichtlich repräsentieren.

Für die Durchführung der definierten Prozesse muss eine durchgängige Verarbeitung von Informationen möglich sein. Alle relevanten Prozess-, Auftrags-, Maschinen- und Nutzungsdaten müssen formal abgebildet und deren Strukturen und Verknüpfungen spezifiziert werden. Die Spezifikation der Datenstrukturen muss die definierten Informationen – die Klassen, Attribute und Methoden sowie die Beziehungen zwischen den Elementen – formal abbilden. Vor allem müssen Informationen zu bearbeitbaren Materialien, vorhandenen Laserbearbeitungsmaschinen und möglichen Verfahren abgebildet werden. Insbesondere die Beziehungen zwischen den Daten müssen zur Abbildung von Technologiedaten und des Nutzungsmanagements im Informationsmodell eindeutig erkennbar sein.

Anforderung 16:

Das Informationsmodell muss die für die Auswahl der geeigneten Laserbearbeitungsmaschinen erforderlichen Informationen und Beziehungsstrukturen abbilden.

Die festgelegten Eingaben geben Auskunft über die Eignung der vorhandenen Laserbearbeitungsmaschinen für den jeweiligen Fertigungsauftrag. Diese Informationen müssen dem Anwender zu Beginn des Fertigungsprozesses bekannt sein. Sie basieren auf verfahrens- und auftragsspezifischen Eingaben und müssen im Informationsmodell abgebildet werden. Der Entscheidungsprozess hinsichtlich der Maschinenauswahl geht auf Grundlage von Informationen vor sich, die durch spezifizierte Verknüpfungen abgebildet werden.

Anforderung 17:

Das Informationsmodell muss die für die Lizenzmodellauswahl erforderlichen Informationen und Beziehungsstrukturen abbilden.

Das Informationsmodell muss umfassend genug sein, um die notwendigen Informationen und ihre Beziehungen darzustellen und dadurch die angemessene Auswahl eines Lizenzmodells zu unterstützen. Auf Basis der vom Anwender eingegebenen Informationen werden die passenden Lizenzmodelle für einen Auftrag gewählt. Diese Informationen müssen in der Planungsphase

des Fertigungsprozesses bekannt sein. Sie beziehen sich auf auftrags- und erfahrungsspezifische Eingaben.

Anforderung 18:

> *Das Informationsmodell muss Informationen zur Prüfung der Datensicherheit abbilden und zur Verfügung stellen.*

Um die Datensicherheit und das Erreichen der Schutzziele zu gewährleisten, werden Informationen benötigt. Nur wenn diese Informationen aus den Zertifikaten der Anwender und der Technologiedatenmarktplätze abgebildet werden, kann die Integrität, Authentizität und Verbindlichkeit der Daten überprüft werden. Außerdem muss das Informationsmodell benötigte Informationen zur Entschlüsselung der Daten beinhalten, um deren Vertraulichkeit zu gewährleisten.

Anforderung 19:

> *Das Informationsmodell muss Informationen aus der Nutzungsphase bezüglich der vereinbarten Nutzungsregeln von erworbenen Technologiedaten integrieren und bereitstellen.*

Anforderung 20:

> *Das Informationsmodell muss Informationen zur Prüfung der Nutzungsregeln abbilden und zur Verfügung stellen.*

Anforderung 21:

> *Das Informationsmodell muss eine klare Zuordnung der Lizenzen und Lizenzmodelle zu den Technologiedaten ermöglichen.*

Für die Sicherstellung einer kontrollierten Nutzung von erworbenen Technologiedaten sind Informationen zu den vereinbarten Nutzungsregeln sowie zum Stand der Nutzung der Daten im Informationsmodell abzubilden. Diese Informationen müssen mit den jeweiligen Lizenzen und Technologiedaten verknüpft werden. Ferner ist eine Zuordnung der Nutzungsregeln zu den jeweiligen Lizenzen erforderlich, um die durchgängige Nachverfolgung der Datennutzung zu

gewährleisten. Die Zugriffsberechtigungen bzw. Nutzungsregeln sowie die Informationen zum Nutzungsstand vor und nach dem Laserbearbeitungsprozess müssen formal abgebildet werden und sie müssen stets zur Verfügung stehen. Die Lizenzen werden dafür zunächst mit einer Identifikationsnummer versehen. Diese ID muss für die weiteren Schritte und zur Verfolgung der Nutzung bei Bedarf zur Verfügung gestellt werden.

3.4.4 Anforderungen an die Implementierung

Unter „Implementierung“ wird die Umsetzung des entwickelten Gesamtkonzepts verstanden. Dazu wird das Assistenzsystem als Softwareapplikation implementiert. Diese Software dient der Bereitstellung, Verwaltung und Verteilung von Technologiedaten und der zugehörigen Lizenzen. Außerdem soll sie den Anwender bei der Maschinen- und Lizenzmodellauswahl unterstützen und die Datensicherheit gewährleisten. Die Anforderungen, welche an die Implementierung gestellt werden, werden im Folgenden vorgestellt:

Anforderung 22:

> *Die Applikation muss dem Anwender eine grafische Benutzungsoberfläche zur Interaktion und Navigation bereitstellen.*

Die Softwareapplikation muss über eine grafische Benutzungsoberfläche verfügen, welche es dem Anwender ermöglicht, die geforderten Funktionalitäten abzurufen. Außerdem muss der Anwender in der Lage sein, Daten über die Applikation einzugeben und von dieser Informationen zu beziehen.

Anforderung 23:

> *Die Applikation muss die Eingabe der für die Maschinenauswahl nötigen Informationen ermöglichen.*

Anforderung 24:

> *Die Applikation muss die Durchführung der Lizenzmodellauswahl ermöglichen.*

Anforderung 25:

Die Applikation muss die Empfehlungen zur Maschinen- und Lizenzmodellauswahl bereitstellen.

Bevor eine endgültige Entscheidung hinsichtlich der verwendeten Laserbearbeitungsmaschine in der Planungsphase gefällt wird, werden unter Bezugnahme auf die festgelegten auftragsrelevanten Informationen die geeigneten Geräte zusammengestellt. Danach werden auf Basis der Eingaben des Anwenders die passenden Lizenzmodelle eruiert. Die Softwareanwendung muss die Eingabe von entsprechenden Informationen ermöglichen. Im Hintergrund muss ein Algorithmus zu deren Verarbeitung ablaufen. Darüber hinaus müssen die Ergebnisse als Handlungsempfehlungen der Auftragsplanung zur Verfügung gestellt werden. Die Softwareanwendung muss also die für den Endentscheidungsprozess erforderlichen Informationen und Werkzeuge bereitstellen.

Anforderung 26:

Die Applikation muss eine Datenschnittstelle für das Abziehen der Technologiedaten und deren Weiterleitung an die Laserbearbeitungsmaschine bereitstellen.

Im Anschluss an den Erwerb werden die Technologiedaten aufbereitet und an den Anwender übermittelt. Diese Daten werden in einem speziell zugewiesenen, freigegebenen Netzwerkordner hinterlegt. Wenn sie benötigt werden, soll die implementierte Applikation in der Lage sein, die Technologiedaten abzurufen und für die jeweilige Laserbearbeitungsmaschine bereitzustellen.

Anforderung 27:

Die Applikation muss eine Datenschnittstelle für die Bereitstellung der Lizenzdaten zur Verfügung stellen.

Die für die Laserbearbeitung benötigten Technologiedaten werden auf den Technologiedatenmarktplätzen von Maschinenherstellern nach verschiedenen Kaufmodellen und Spezifikationen angeboten. Nach dem Erwerben wird eine Lizenz entsprechend der zwischen dem Anwender und dem Technologiedatenmarktplatz vereinbarten Nutzungsregeln erstellt, die dem Anwender zur Verfügung gestellt wird. Die Softwareapplikation muss eine Datenschnittstelle bereitstellen, die dem Bezug der Lizenzen dienlich ist. Über diese Schnittstelle werden Lizenzanfragen versendet und Entschlüsselungsinformationen empfangen.

Anforderung 28:

Die Applikation muss die Informationen zum Nutzungszustand darstellen können.

Die Nachverfolgung der Technologiedatennutzung ist eine essentielle Aufgabe der Softwareapplikation. Die entsprechenden Informationen müssen aktualisiert und zurückgeführt werden, damit sie bei Bedarf vom Anwender abgerufen werden können.

Anforderung 29:

Die Applikation muss die Authentifizierung des Benutzers ermöglichen.

Die erworbenen Technologiedaten dürfen nur von autorisierten Benutzern verwendet werden. Daher muss sich der Benutzer zunächst authentifizieren, bevor er die Lizenzdaten abrufen kann. Die Applikation muss dementsprechend eine Möglichkeit bieten, dass der Nutzer seine persönlichen Daten eingeben kann, um sich auszuweisen. Dadurch werden die Risiken zum unautorisierte Nutzung der Daten unterbindet.

3.4.5 Zusammenfassung der Anforderungen

Das Anforderungsprofil setzt sich aus Anforderungen an die Methode bei der Konzeption, an das Informationsmodell und an die Implementierung des Gesamtkonzeptes zusammen. Es wurden insgesamt 29 Anforderungen identifiziert. Das erarbeitete Anforderungsprofil wurde durch die Art der jeweiligen Anforderung nach [149] ergänzt. Es wird dahingehend zwischen Festforderungen (F) und Wünschen (W) differenziert. Festforderungen *müssen* im Verlauf der Konzeptionierung und Implementierung berücksichtigt werden. Im Gegensatz dazu *soll* auf Wünsche eingegangen werden, sie müssen jedoch nicht unbedingt erfüllt werden. Das dargestellte Anforderungsprofil dient als Grundlage für die Entwicklung und Implementierung des Konzepts. Zudem soll es dabei helfen, festzustellen, ob die Gesamtkonzeption letztlich den Anforderungen entspricht.

In Tabelle 3-1 werden die Anforderungen an die Methode zusammengefasst. Tabelle 3-2 stellt die Anforderungen an das Informationsmodell und Tabelle 3-3 die Anforderungen an die Implementierung dar.

Tabelle 3-1: Anforderungsprofil bezüglich der Methode

Nr.	Art	Bezeichnung
Anforderungen an die Methode		
1	F	Die Methode muss eine Vorgehensweise definieren, anhand der ein auftragsspezifischer Ansatz zur Auswahl der geeigneten Laserbearbeitungsmaschinen entwickelt werden kann.
2	F	Die Methode muss eine Vorgehensweise definieren, anhand der eine rechnerbasierte Maschinenauswahl ermöglicht wird.
3	F	Die Methode muss eine Vorgehensweise definieren, anhand der ein effektiver und rechnerbasierter Ansatz zur Auswahl der passenden Lizenzmodelle realisiert werden kann.
4	F	Die Methode muss eine Vorgehensweise definieren, anhand der eine gezielte Zuteilung der Daten zu den Laserbearbeitungsmaschinen möglich ist.
Sicherheitsanforderungen an der Methode		
5	F	Die Methode muss eine Vorgehensweise definieren, anhand der die Vertraulichkeit der verarbeitenden Daten gewährleistet werden kann.
6	F	Die Methode muss eine Vorgehensweise definieren, anhand der die Integrität der Daten sichergestellt wird.

7	F	Die Methode muss eine Vorgehensweise definieren, welche die Authentifizierung der Kommunikationspartner ermöglicht.
8	F	Die Methode muss eine Vorgehensweise definieren, anhand der eine Überprüfung der Nutzungsregeln vor dem Laserbearbeitungsprozess sichergestellt wird.
9	F	Die Methode muss eine Vorgehensweise definieren, die die Nachverfolgung von Änderungen während der Datenverarbeitung ermöglicht.
10	F	Die Methode muss eine Vorgehensweise definieren, anhand der die Verfügbarkeit von Daten sichergestellt wird.
11	F	Die Methode muss eine Vorgehensweise definieren, anhand der eine Überprüfung der Verbindlichkeit bzw. Nicht- Abstreitbarkeit sichergestellt wird.
12	W	Die Methode soll eine Vorgehensweise definieren, anhand der eine Autorisierungsstrategie zur Verwaltung der Zugriffsrechte von Nutzern erarbeitet wird.

Tabelle 3-2: Anforderungsprofil bezüglich des Informationsmodells

Nr.	**Art**	**Bezeichnung**
Anforderungen an das Informationsmodell		
13	F	Das Informationsmodell muss maschinen-, material-, auftrags- und prozessrelevante Informationen enthalten.

14	F	Das Informationsmodell muss die Datenstrukturen und -beziehungen formal abbilden.
15	F	Das Informationsmodell muss die Verknüpfungen zwischen den Datenstrukturen übersichtlich repräsentieren.
16	F	Das Informationsmodell muss die für die Auswahl der geeigneten Laserbearbeitungsmaschinen erforderlichen Informationen und Beziehungsstrukturen abbilden.
17	F	Das Informationsmodell muss die für die Lizenzmodellauswahl erforderlichen Informationen und Beziehungsstrukturen abbilden.
18	F	Das Informationsmodell muss Informationen zur Prüfung der Datensicherheit abbilden und zur Verfügung stellen.
19	F	Das Informationsmodell muss Informationen aus der Nutzungsphase bezüglich der vereinbarten Nutzungsregeln von erworbenen Technologiedaten integrieren und bereitstellen.
20	F	Das Informationsmodell muss Informationen zur Prüfung der Nutzungsregeln abbilden und zur Verfügung stellen.
21	F	Das Informationsmodell muss eine klare Zuordnung der Lizenzen und Lizenzmodelle zu den Technologiedaten ermöglichen.

Tabelle 3-3: Anforderungsprofil bezüglich der Implementierung

Nr.	Art	Bezeichnung
Anforderungen an die Implementierung		
22	F	Die Applikation muss dem Anwender eine grafische Benutzungsoberfläche zur Interaktion und Navigation bereitstellen.
23	F	Die Applikation muss die Eingabe der für die Maschinenauswahl nötigen Informationen ermöglichen.
24	F	Die Applikation muss die Durchführung der Lizenzmodellauswahl ermöglichen.
25	F	Die Applikation muss die Empfehlungen zur Maschinen- und Lizenzmodellauswahl bereitstellen.
26	F	Die Applikation muss eine Datenschnittstelle für das Abziehen der Technologiedaten und deren Weiterleitung an die Laserbearbeitungsmaschine bereitstellen.
27	F	Die Applikation muss eine Datenschnittstelle für die Bereitstellung der Lizenzen zur Verfügung stellen.
28	F	Die Applikation muss die Informationen zum Nutzungszustand darstellen können.
29	F	Die Applikation muss die Authentifizierung des Benutzers ermöglichen.

4 KONZEPT

Die Ausführungen in diesem Kapitel beruhen auf dem bereits dargelegten Stand der Technik (siehe Kapitel 2). Im Sinne der im vorherigen Kapitel festgelegten Ziele und der daraus abgeleiteten Anforderungen wird im Folgenden ein Konzept für die Unterstützung des Anwenders bei der effektiven und sicheren Nutzung von Technologiedatenmarktplätzen am Beispiel der Lasermarkierung entwickelt. Dieses Konzept bildet die Grundlage für die prototypische Implementierung eines entsprechenden Assistenzsystems als eine Softwareapplikation (siehe Kapitel 5). Die Entwicklung dieses System beruht auf Methoden der Informationsmodellierung, Entscheidungsfindung, IT- Sicherheit und künstlichen Intelligenz. Letztlich soll das entwickelte System Handlungsempfehlungen liefern für die Auswahl von Laserbearbeitungsmaschinen und von Lizenzmodellen, die für den Erwerb der Technologiedaten auf Technologiedatenmarktplätzen benötigt werden. Außerdem bietet das System einen vertrauenswürdigen Ansatz zur sicheren und übersichtlichen Nutzung von erworbenen Technologiedaten. Die Merkmale des Assistenzsystems werden nachfolgend spezifiziert und modelliert. Ferner wird dargelegt, welche Informationen die Nutzer bereitstellen müssen und welche durch das System generiert werden. Abbildung 4-1 gibt eine Übersicht über dieses Kapitel.

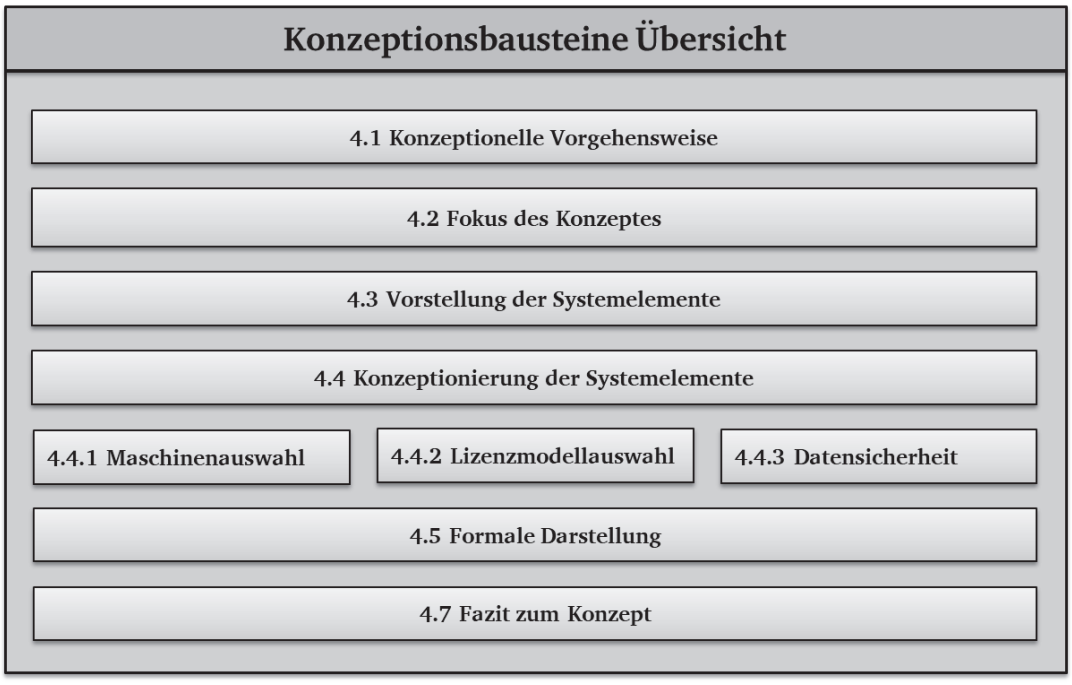

Abbildung 4-1: Übersicht – Kapitel 4 (eigene Darstellung)

Im folgenden Kapitel werden zunächst die für Entwicklung des Konzeptes angewandten Methoden vorgestellt. Danach wird der Fokus des Konzeptes hinsichtlich der Planungs- und der Produktionsphase in der Laserbearbeitung dargestellt. Es folgt die Vorstellung der Systemelemente Maschinenauswahl, Lizenzmodellauswahl und Datensicherheit. Danach wird dargelegt, wie diese Systemelemente konzipiert und modelliert wurden. Im Unterkapitel 4.5 wird die Entwicklung der Konzeption und UML-Klassendiagramme zur Unterstützung des Anwenders bei der effektiven Bereitstellung und sicheren Nutzung von Technologiedaten in der Lasermarkierung vorgestellt. Und zuletzt wird ein Fazit zum entwickelten System gegeben. Nun werden die Methoden und Werkzeuge zur Entwicklung des Konzeptes vorgestellt.

4.1 Konzeptionelle Vorgehensweise

Das Hauptaugenmerk der vorliegenden Dissertation liegt auf der Entwicklung eines Konzeptes zur Unterstützung des Anwenders bei der effektiven und sicheren Nutzung von Technologiedatenmarktplätzen. Am Beispiel der Lasermarkierung wird gezeigt, wie das rechnerunterstützte Assistenzsystem in Gestalt einer Softwareapplikation Handlungsempfehlungen für die Planungsphase hinsichtlich der Auswahl des Maschinen- und des Lizenzmodells liefert. Ferner soll es eine Lösung bieten für den Schutz von erworbenen Technologiedaten. Es wird demnach für die Industrie ein effizientes, flexibles und sicheres Konzept zur Nutzung von Technologiedatenmarktplätzen entwickelt. Der Entwicklung des vorliegenden Konzeptes liegt ein modellbasiertes Vorgehen zugrunde. Hierfür werden sämtliche Aktivitäten der Nutzer identifiziert, beginnend mit dem Starten des Assistenzsystems bis hin zum Senden der Technologiedaten an die jeweilige Laserbearbeitungsmaschine. Die Informationsmodellierung ermöglicht es, ein unmissverständliches und implementierbares Konzept zu erstellen. Solche Modellierungsansätze finden zunehmend Anwendung, weil dank ihnen die steigende Komplexität beherrscht und eine einheitliche Grundlage für die Entwicklung von Softwarelösungen geschaffen werden kann [120].

Zunächst wird dargelegt, welche Informationen die Nutzer bereitstellen müssen, damit das Assistenzsystem die benötigten Daten liefern kann. Diese Informationen werden für die Informationsmodellierung formalisiert. Ferner gilt es, Kommunikations- und Datenflüsse zwischen Anwendern und Technologiedatenmarktplätzen zu beschreiben, um darlegen zu können, wie das Sicherheitskonzept in das Informationsmodell integriert werden kann. Meine konzeptionelle Vorgehensweise zur Bereitstellung einer rechnerunterstützten Maschinenauswahl umfasst zudem Entscheidungsbäume sowie Operationen aus der

Mengentheorie. Für die Konzipierung der Auswahl von Lizenzmodellen wird auf Methoden aus dem Bereich der künstlichen Intelligenz zurückgegriffen, speziell auf die Anwendung von Assistenzsystemen.

Um den bestehenden Anforderungen an die Datensicherheit gerecht zu werden, müssen darüber hinaus etablierte Sicherheitstechnologien und kryptografische Methoden in das Konzept integriert werden. Zur Lizenzierung von Technologiedaten werden besonders fortschrittliche Technologien wie Blockchain und Smart Contracts verwendet. Mit der vorliegenden Konzeption für die Nutzung von Technologiedatenmarktplätzen sollen die Forschungsfragen aus dem Kapitel 1.1 beantwortet werden.

Mithilfe der Informationsmodelle (siehe Kapitel 4.5) können Informationen identifiziert, strukturiert und formalisiert werden. Zur Abbildung der Informationen und der Beziehungen zwischen den Systemelementen dienen hier die UML2-Diagramme (engl. Unified Modeling Language – siehe Kapitel 2). Basierend auf den vorgestellten Konzeptionsbausteinen sowie dem entwickelten Informationsmodell werden dem Anwender Handlungsempfehlungen zur Maschinen- und Lizenzmodellauswahl und zur Datensicherheit bereitgestellt. Die Konzeption und Entwicklung des Assistenzsystems erfolgt, wie im vorherigen Unterkapitel dargelegt wurde (siehe Abbildung 4-1), in mehreren Schritten unter Verwendung der im Unterkapitel 4.4 vorgestellten Konzeptionsbausteine.

4.2 Fokus des Konzeptes

Das in dieser Dissertation entwickelte Konzept soll hier beispielhaft Anwendern von Laserbearbeitungsmaschinen verschiedener Hersteller zugutekommen. In Abbildung 4-2 wird der Fokus des Konzeptes dargestellt. Berücksichtigt wurden die beteiligten Systeme und Datenflüsse sowie die Eingangs- und Ausgangsinformationen in der Planungs- und Produktionsphase der Lasermarkierung.

Für die Lasermarkierung werden für jede Laserbearbeitungsmaschine spezifische Technologiedaten benötigt, die Informationen zu den Maschineneinstellungen beinhalten. Die Technologiedaten unterscheiden sich demnach je nach Maschinen- und Materialeigenschaften und Lasertechnologie. Wird einem Anwender ein Auftrag für einen bestimmten Laserbearbeitungsprozess erteilt und hat er keine passenden Technologiedaten dafür, so muss er sich auf Basis verfügbarer Informationen für eine Laserbearbeitungsmaschine entscheiden

und dann die benötigten Technologiedaten unter Anwendung eines Lizenzmodells auf dem entsprechenden Technologiedatenmarktplatz erwerben.

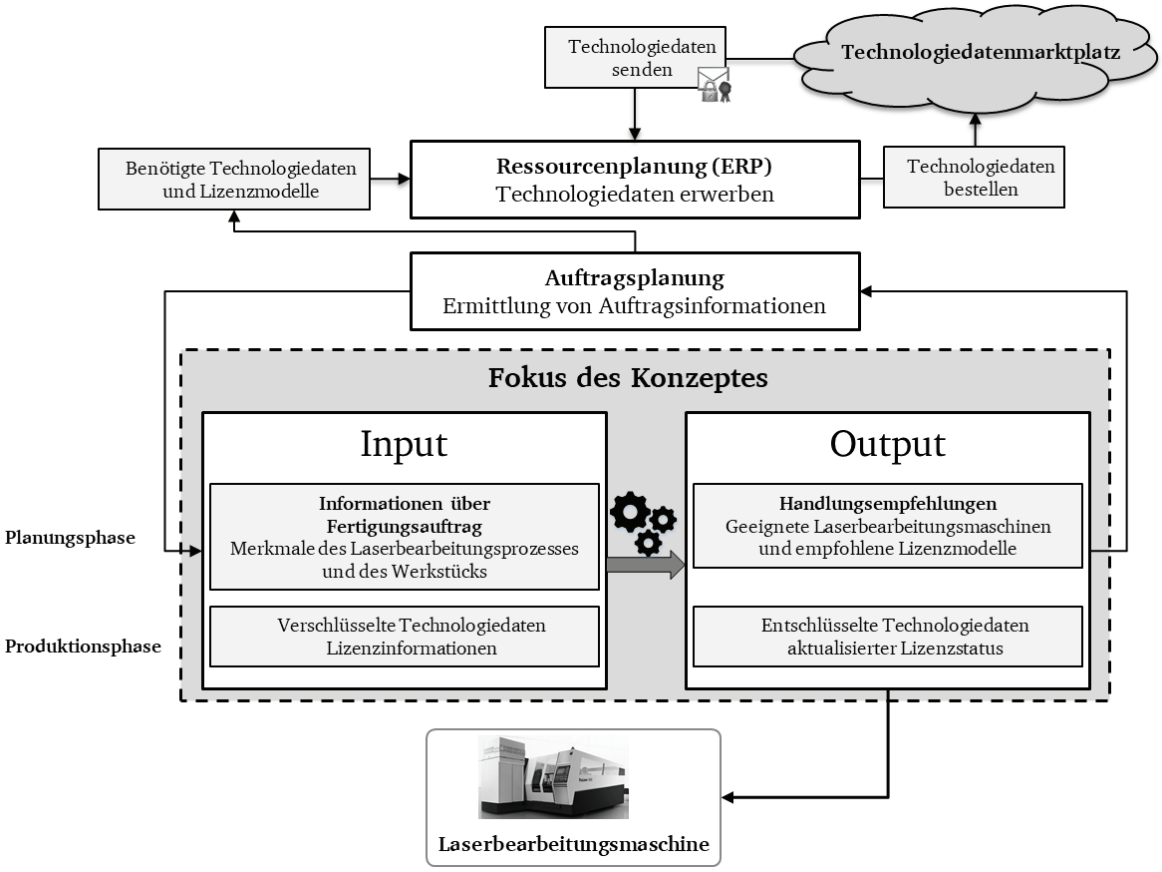

Abbildung 4-2: Fokus des Konzeptes in der Laserbearbeitung (eigene Darstellung)

Nach dem Kauf der Technologiedaten werden diese aufbereitet und mitsamt der Lizenz dem Anwender bereitgestellt. Eine ausführliche Beschreibung der Funktionsweise der Online-Technologiedatenmarktplätze erfolgte in Kapitel 2.2.4. Die Maschinenhersteller betreiben je eigene Marktplätze, auf denen sie die von ihnen entwickelten Technologiedaten unter verschiedenen Bedingungen und verknüpft mit unterschiedlichen Nutzungsbedingungen feilbieten. Bei den Daten wird zwischen dem Typ der Laserbearbeitungsmaschine, den Werkstückmerkmalen und den Prozessbedingungen unterschieden.

Technologiedaten werden im Rahmen eines Nutzungsmodells erworben. Konkret werden Lizenzen gekauft, die beispielweise für eine bestimmte Zeitspanne, für eine beschränkte Anzahl von Anwendungen oder auch für bestimmte Orte oder Maschinen Gültigkeit haben. Der Erwerb von Technologiedaten wird im Rahmen der Ressourcenplanung vollzogen. Nachdem eine Zahlung eingegangen ist, wird eine Lizenz für die erworbenen Technologiedaten erteilt. Die

Lizenz beinhaltet die vereinbarten Nutzungsregeln und die für die Nutzung der Technologiedaten erforderlichen Informationen. Schließlich werden die Technologiedaten vorbereitet, mittels standarisierter Verfahren verschlüsselt und dem Anwender über eine sichere Netzwerkverbindung übermittelt. Dafür müssen vorab Informationen über die Technologiedaten und über das gewünschte Lizenzmodell vorliegen, die im Rahmen der Auftragsplanung gesammelt werden.

In der Planungsphase wird entschieden, auf welcher Laserbearbeitungsmaschine ein Auftrag bearbeitet wird und welche Ressourcen dafür mobilisiert werden müssen. Im Konzept liegt der Fokus auf der effektiven Bereitstellung der benötigten Technologiedaten. Für deren Bezug bedarf es Informationen über den Fertigungsauftrag sowie über die Merkmale des Laserbearbeitungsprozesses und des Werkstücks. Diese Informationen werden bearbeitet, um Handlungsempfehlungen hinsichtlich der geeigneten Laserbearbeitungsmaschine und der empfohlenen Lizenzmodelle bereitzustellen. Diese Handlungsempfehlungen werden dann an die mit der Auftragsplanung betrauten Personen übermittelt, damit diese angemessene Entscheidungen treffen können. Um einen Laserbearbeitungsauftrag abzuarbeiten, bedarf es entsprechender Maschinen, an die passende Technologiedaten übermittelt werden müssen. In der Produktionsphase müssen erworbenen Technologiedaten an der jeweilige Laserbearbeitungsmaschine zur Verfügung gestellt werden. Das im Rahmen dieser Dissertation entwickelte Assistenzsystem ermöglicht einen sicheren Datentransfer, was auch die Verwaltung von erworbenen Technologiedaten und dazugehöriger Lizenzen miteinschließt. Dazu gehört der Lizenzabruf, die Entschlüsselung der Technologiedaten auf Empfängerseite und das Weiterleiten an die jeweilige Laserbearbeitungsmaschine. Nach der Verwendung der Technologiedaten werden die Nutzungsregeln der Lizenz aktualisiert. Sollte die Lizenz nun ungültig sein, können die Technologiedaten nicht erneut entschlüsselt werden, weshalb ggf. neue bezogen werden müssen.

Es gibt bisher keine systematische Vorgehensweise und keine Handlungsempfehlungen bezüglich der effektiven Bereitstellung von Technologiedaten. Bei der Auswahl geeigneter Laserbearbeitungsmaschinen oder des passenden Lizenzmodells wareE2En die Anwender bislang auf sich gestellt. Außerdem wurden die erworbenen Technologiedaten und die zugehörigen Lizenzen bisher nicht hinsichtlich der Datensicherheit untersucht. Das entwickelte Konzept deckt diese Forschungslücke ab. In dieser Dissertation wird ein sicheres Assistenzsystem entwickelt, das dem Anwender Handlungsempfehlungen zur effektiven und sicheren Nutzung von Technologiedatenmarktplätzen liefert. Das konzipierte System

unterstützt den Anwender bei der Auswahl der Maschinen und des Lizenzmodells und ermöglicht eine sichere Übermittlung und Nutzung der Technologiedaten.

4.3 Vorstellung der Systemelemente

In diesem Kapitel werden die entwickelten Systemelemente vorgestellt und beschrieben. Diese Elemente repräsentieren die Funktionalitäten des Systems und bilden grundsätzlich dessen Struktur ab (siehe Abbildung 4-3). Ferner werden Interaktionen zwischen den Systemelementen und den beteiligten Systemen auf untenstehender Abbildung dargestellt.

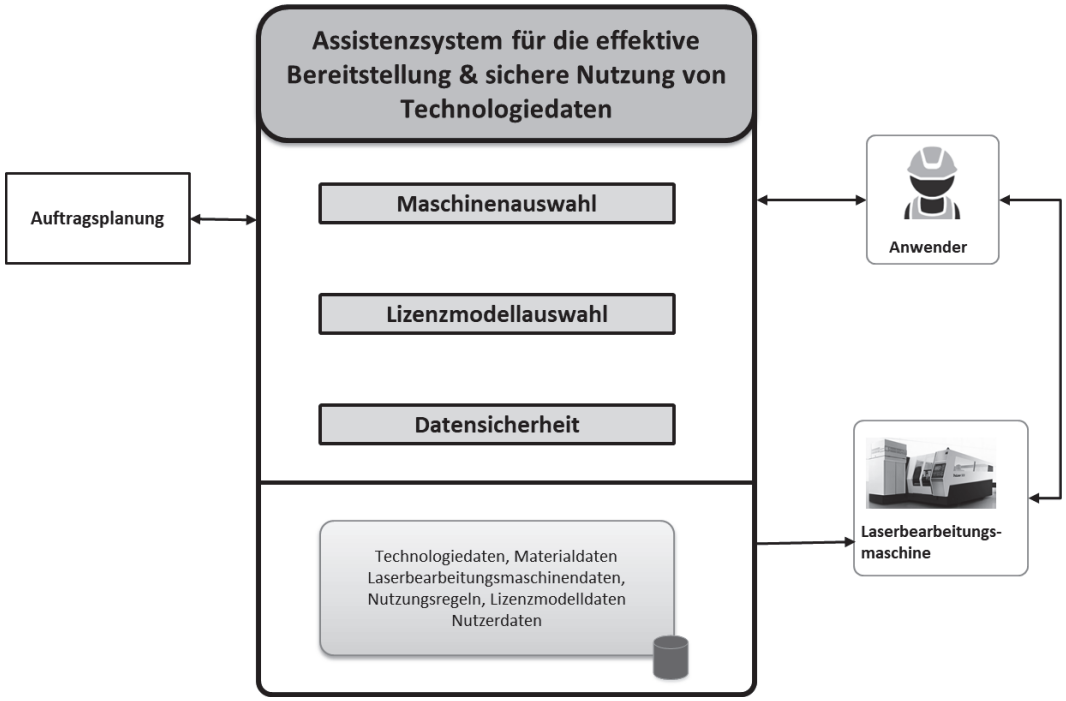

Abbildung 4-3: Darstellung der Systemelemente (eigene Darstellung)

Die Systemelemente unterstützen den Anwender bei der effektiven und sicheren Nutzung von Technologiedatenmarktplätzen und unterbreiten Handlungsempfehlungen hinsichtlich der Auswahl der Laserbearbeitungsmaschine und des Lizenzmodells sowie bezüglich der Gewährleistung der Datensicherheit.

Das Element „Maschinenauswahl“ unterstützt den Anwender bei der Beantwortung der Frage, welche Laserbearbeitungsmaschine zur Fertigung eines Auftrags verwendet werden soll. Diese Frage wird bei der Konzeption als Auswahlproblem betrachtet. Das Ziel besteht darin, über die

Eignung von vorhandenen Laserbearbeitungsmaschinen für einen bestimmten Auftrag zu bescheiden. Die Auswahl der Maschine erfolgt nach Kriterien, die auf Basis der Auftragsinformationen sowie unter Bezugnahme auf die Eigenschaften der Laserbearbeitungsmaschinen festgelegt werden. Das erste Kriterium ist die Art des Laserbearbeitungsprozesses – handelt es sich um einen Laserschneidprozess, einen Laserschweißprozess oder einen Lasermarkierprozess? Das zweite Kriterium bezieht sich auf die benötigte Technologie, also darauf, ob zweidimensionale oder dreidimensionale Werkstücke oder Rohrwerkstücke ausgearbeitet werden. Das dritte Kriterium besteht in der Materialart und der Materialdicke des Werkstücks. Hierbei wird überprüft, ob die im Auftrag genannte Materialart in der Materialliste einer Laserbearbeitungsmaschine auftaucht und ob die Materialdicke des Werkstücks im Bearbeitungsbereich der Maschine liegt. Diese drei Kriterien spielen bei der Auswahl der Laserbearbeitungsmaschine eine entscheidende Rolle. Das Ergebnis dieses Prozesses ist eine Menge an für einen Auftrag geeigneten Laserbearbeitungsmaschinen. Die Entscheidung, auf welcher dieser Maschinen der Auftrag letztlich ausgeführt wird, fällt im Rahmen der Auftragsplanung und ist daher nicht Gegenstand dieser Dissertation. Die Konzipierung der Maschinenauswahl wird in Kapitel 4.4.1 detailliert dargestellt.

Das Element „Lizenzmodellauswahl“ beinhaltet Handlungsempfehlungen für den Anwender hinsichtlich der Frage, wie die Technologiedaten bereitgestellt werden können, wie also die Auswahl des passenden Lizenzmodells ablaufen kann. Hierbei werden Lizenzmodelle vorgestellt, die beim Erwerben von Technologiedaten in Frage kommen können. Basierend auf den Ausführungen in Kapitel 2.4.5 werden beispielhaft Lizenzmodelle untersucht, wobei zentrale Merkmale herausgearbeitet werden. Hierbei werden die Art, der Typ und die Metrik einer Lizenz als Kriterien für die Auswahl festgelegt. Schließlich wird das schrittweise Vorgehen zur flexiblen und effektiven Auswahl eines geeigneten Lizenzmodells anhand eines Assistenzsystems dargestellt. Das Assistenzsystem verfügt über eine Wissensbasis (engl. Knowledge Base), die Informationen von menschlichen Experten enthält. Diese Informationen werden in einer Inferenzmaschine (engl. Inference Engine) verarbeitet und die daraus abgeleiteten Ergebnisse werden dem Benutzer über eine Benutzerschnittstelle (engl. User Interface) zugänglich gemacht.

Das dritte Systemelement betrifft die „Datensicherheit“, bezogen auf den Schutz von Technologiedaten vom Erwerb bis zur Nutzung auf der jeweiligen Laserbearbeitungsmaschine. Hierbei liegt das Augenmerk auf der Sicherstellung der in Kapitel 2.4.1 beschriebenen Schutzziele. Dazu bedarf es der Beschreibung eines durchgängigen Lizenzierungskonzeptes, und zwar sowohl hinsichtlich der Vorgänge zur Vorbereitung der Technologiedaten als auch

jener zu deren Nutzung bei den Anwendern. Außerdem werden die verschiedenen Auslieferungsmethoden einer Lizenz untersucht und bewertet. Ferner werden benötigte kryptografische Vorgehensweisen vorgestellt und in das Gesamtkonzept integriert.

Die drei Konzeptbausteine werden in den folgenden Kapiteln 4.4.1–4.4.3 detailliert beschrieben. Basierend auf diesen Bausteinen wurde ein sicheres Assistenzsystem entwickelt, das Handlungsempfehlungen zur Maschinen- und Lizenzmodellauswahl sowie hinsichtlich der Datensicherheit liefert.

4.4 Konzeptionierung der Systemelemente

Die hier beschriebene Konzeption des entwickelten Assistenzsystems umfasst im Kern drei Hauptbausteine. Der erste Konzeptbaustein, die Maschinenauswahl, besteht in der Entwicklung eines Tools, dank dessen der Anwender eine für seine Zwecke geeignete Laserbearbeitungsmaschine auswählen kann. In diesem Zusammenhang werden zunächst sinnvolle Auswahlkriterien identifiziert und beschrieben. Danach wird die methodische Vorgehensweise bei der Maschinenauswahl vorgestellt. Der zweite Konzeptbaustein umfasst die Modellierung der Lizenzmodellauswahl. Hierbei werden zentrale Merkmale von Lizenzmodellen, die zur Auswahl stehen, beschrieben. Und schließlich wird das Assistenzsystem zur Bereitstellung von Handlungsempfehlungen für die Auswahl der am besten geeigneten Lizenzmodelle dargelegt und formalisiert. Der dritte Konzeptbaustein betrifft die Datensicherheit. In diesem Kontext werden Möglichkeiten der Lizenzierung sowie der Auslieferung von Lizenzen diskutiert. Dann wird das Lizenzierungskonzept basierend auf Blockchain- und Smart-Contract-Technologien vorgestellt. Schließlich werden kryptografische Operationen vorgestellt, wobei auch berücksichtigt wird, welchen Anforderungen sie bezüglich der Datensicherheit genügen.

4.4.1 Maschinenauswahl

Im Folgenden wird das Systemelement „Maschinenauswahl“ konzipiert. Zur Darstellung des Auswahlprozesses werden Ja/Nein-Entscheidungsbäume verwendet, die letztendlich zur Erstellung eines Klassifikationssystems führen. Bei jedem Verzweigungsknoten wird der Merkmalwert für das festgelegte Attribut abgefragt. Jeder Pfad, von der Wurzel bis zu den letzten Entscheidungsknoten, entspricht einer der Entscheidungsregeln, nach denen eine Klassifizierung der Gesamtmenge aller möglichen Entscheidungen erfolgt. Diese gestuften

Entscheidungsregeln werden in Gestalt einer gerichteten Baumstruktur darstellt. Das bedeutet, dass die Entscheidungen in einer bestimmten Reihenfolge getroffen werden, beginnend an der Wurzel und endend an den Blattknoten [175]–[178]. Diese Struktur ermöglicht es, komplexe Entscheidungsprozesse auf eine übersichtliche und intuitiv verständliche Weise darzustellen.

Die verfügbaren Laserbearbeitungsmaschinen bilden die Gesamtmenge der Maschinen ab, die danach unterschieden werden, ob sie für die Abarbeitung des bestehenden Fertigungsauftrags geeignet sind. Jede Entscheidungsregel resultiert in einer Ja/Nein-Antwort, wobei hier nur die Ja-Entscheidungen berücksichtigt werden, da nur die Menge der geeigneten Laserbearbeitungsmaschinen für das Konzept signifikant ist. Die Knoten des Baums bilden die Attribute, auch „Split-Kriterien“ genannt, ab. Anhand dieser Attribute werden die Laserbearbeitungsmaschinen beschrieben. Sie dienen letztlich dazu, die Maschinen in Untermengen aufzuteilen, bis schließlich nur noch die geeigneten Laserbearbeitungsmaschinen übrigbleiben.

Die Wurzel des Entscheidungsbaums wird von der Anzahl n aller dem Anwender zur Verfügung stehender Laserbearbeitungsmaschinen gebildet.

$$M := \{M_1, M_2, M_3, \dots M_n\} \; \forall \; n \in \mathbb{N} \qquad 4.1$$

Abbildung 4-4 zeigt den Entscheidungsbaum zur Maschinenauswahl inklusive der Entscheidungsregeln.

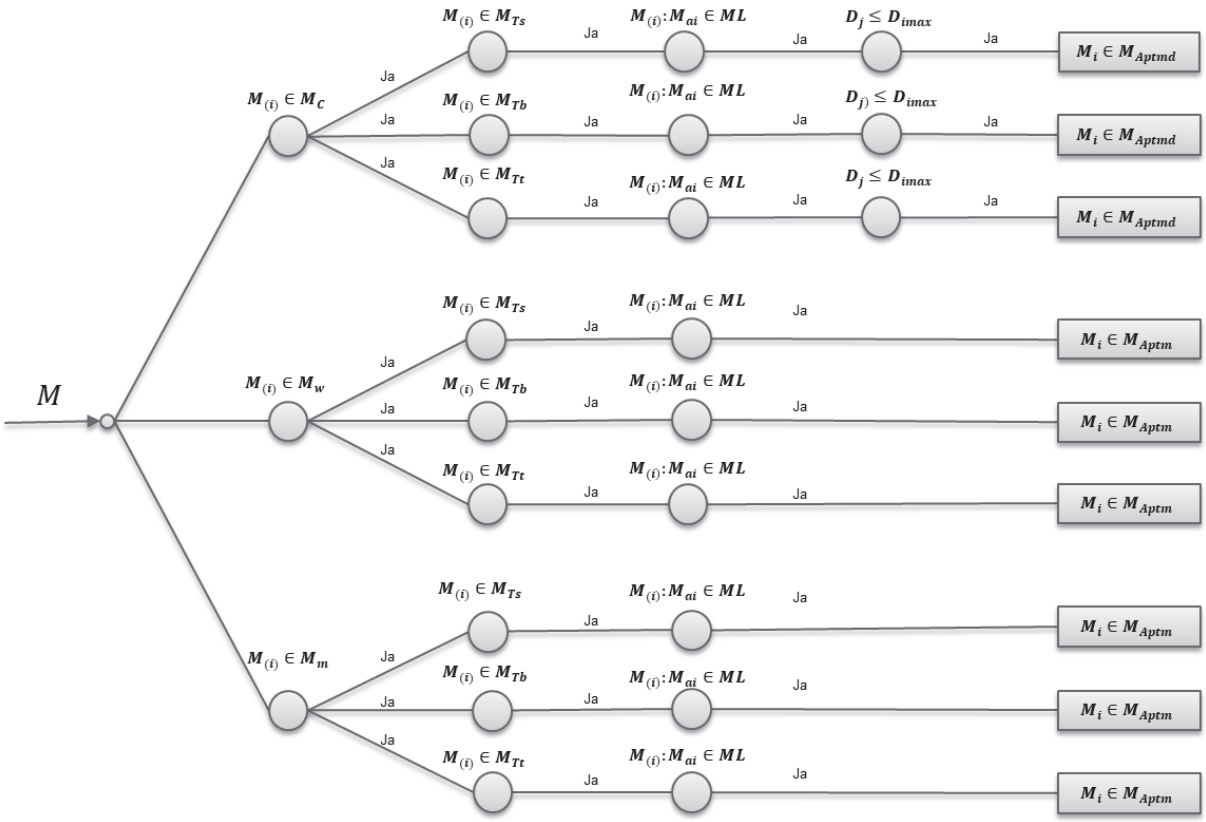

Abbildung 4-4: Entscheidungsbaum zur Maschinenauswahl (eigene Darstellung)

Die ersten drei Knoten bilden die Merkmalwerte des ersten Attributs, der Art des Laserbearbeitungsprozesses, ab. Im Konzept werden drei Laserbearbeitungsprozesse betrachtet. Analog zu diesen Prozessen wird unter den Laserbearbeitungsmaschinen zwischen Schneid-, Schweiß- und Markierlaserbearbeitungsmaschinen unterschieden. Entsprechend teilt sich der Entscheidungsbaum in drei Zweige auf. Die dabei entstehenden Mengen werden im Folgenden formal dargestellt:

Menge aller dem Anwender zur Verfügung stehenden Schneidlaserbearbeitungsmaschinen (engl. Cutting Laser Machines) mit der Anzahl o:

$$M_c := \{M_{c1}, M_{c2}, M_{c3}, \dots M_{co}\}\ \forall o \in \mathbb{N} \qquad 4.\ 2$$

Menge aller dem Anwender zur Verfügung stehenden Schweißlaserbearbeitungsmaschinen (engl. Weld Laser Machines) mit der Anzahl p:

$$M_w := \{M_{w1}, M_{w2}, M_{w3}, \dots M_{wp}\}\ \forall p \in \mathbb{N} \qquad 4.\ 3$$

Menge aller dem Anwender zur Verfügung stehenden Markierlaserbearbeitungsmaschinen (engl. Marking Laser Machines) mit der Anzahl q:

$$M_m := \{M_{m1}, M_{m2}, M_{m3}, \dots M_{mq}\} \, \forall q \in \mathbb{N} \qquad 4.\ 4$$

Die drei gebildeten Teilmengen ergeben in Summe die Menge aller dem Anwender zur Verfügung stehenden Laserbearbeitungsmaschinen. Die Menge der Laserbearbeitungsmaschinen, die ein bestimmtes Merkmal aufweisen, ist immer eine Teilmenge oder gleich der Menge aller dem Anwender zur Verfügung stehenden Laserbearbeitungsmaschinen:

$$M := \{M_c \cup M_w \cup M_m\} \, \forall M_c \subseteq M \; und \; M_w \subseteq M \; und \; M_m \subseteq M \qquad 4.\ 5$$

Unter Bezugnahme auf die Auftragsinformationen wird entschieden, welche Art von Laserbearbeitungsmaschine benötigt wird. Im Anschluss wird bei jedem der drei Knoten anhand der jeweiligen Entscheidungsregel überprüft, ob eine Laserbearbeitungsmaschine aus der Menge aller vorhandenen Laserbearbeitungsmaschinen zur Menge der Schneidlaserbearbeitungsmaschinen, der Schweißlaserbearbeitungsmaschinen oder der Markierlaserbearbeitungsmaschinen gehört. Wenn die entsprechende Entscheidungsregel der geeigneten Laserbearbeitungsmaschinenmenge für eine Maschine die Antwort *Ja* ergibt, dann wird diese Maschine in die Menge eingefügt und beim nächsten Knoten bzw. bei der nächsten Entscheidungsregel weiter betrachtet. Die formale Beschreibung der Entscheidungsregeln bei den ersten drei Knoten lautet folgendermaßen:

Für den ersten Knoten: $if \, M_{(i)} \in M_c \; dann \; M_{(i)} \in M_{(A)}$ *4. 6*

Für den zweiten Knoten: $if \, M_{(i)} \in M_w \; dann \; M_{(i)} \in M_{(A)}$ *4. 7*

Für den dritten Knoten: $if \, M_{(i)} \in M_m \; dann \; M_{(i)} \in M_{(A)}$ *4. 8*

Um ein Beispiel zu nennen: Wenn die Menge der für den Prozess *Laserschneiden* geeigneten Laserbearbeitungsmaschinen gesucht wird, dann wird der obere Pfad ausgehend von der Wurzel bis zum ersten Blätterknoten mit den Merkmal *Schneiden* für das Attribut *Laserprozess* verfolgt. Dabei wird die erste Entscheidungsregel überprüft, und es werden nur Laserbearbeitungsmaschinen, bei denen die Antwort *ja* lautet, in die Menge eingefügt und beim nächsten Knoten weiter betrachtet. Alle Laserbearbeitungsmaschinen mit der Antwort *nein*

werden hingegen nicht mehr bei den nächsten Knoten betrachtet: Es gilt hier zudem, dass die Teilmenge der für einen bestimmten Prozess geeigneten Laserbearbeitungsmaschinen ein Teil der Menge aller beim Anwender existierenden Laserbearbeitungsmaschinen ist.

$$M_{Ap} \subseteq M \qquad 4.\ 9$$

Beim zweiten und dritten Knoten, die zur Bildung der Teilmengen der Schweiß- und Gravierlaserbearbeitungsmaschinen dienen, werden die Laserbearbeitungsmaschinen in diesem Fall nicht betrachtet; die entsprechenden Entscheidungsregeln werden deshalb auch nicht überprüft. Das bedeutet, es wird hier nur der erste obere Teilbaum für die nächsten Entscheidungsregeln der Attribute weiterverfolgt (siehe Abbildung 4-5).

Das zweite Attribut ist die Technologieart. Diese wird auf Basis der Werkstückform festgelegt. Im vorliegenden Konzept werden drei Arten von Technologien berücksichtigt: Technologien für die Laserbearbeitung von zweidimensionalen Werkstücken (engl. Sheet) $\{T_s\}$, von dreidimensionalen Werkstücken (engl. Body) $\{T_b\}$ und von Rohrwerkstücken (engl. Tube)$\{T_t\}$.

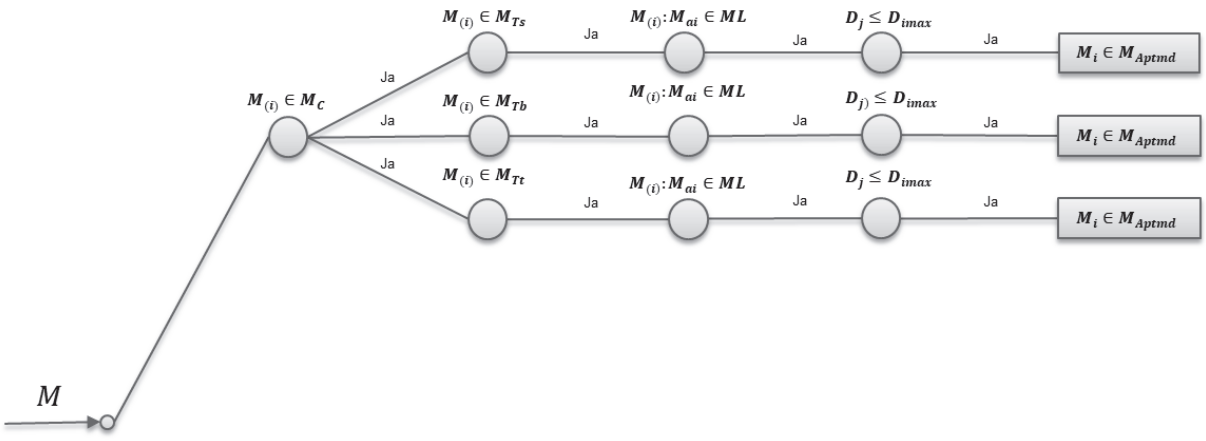

Abbildung 4-5: Entscheidungsbaum zur Maschinenauswahl für Laserschneidprozesse (eigene Darstellung)

Die Menge aller dem Anwender zur Verfügung stehenden Laserbearbeitungsmaschinen, die für die Bearbeitung von zweidimensionalen Werkstücken geeignet sind, lässt sich folgendermaßen darstellen:

$$M_{Ts} := \{M_{Ts1}, M_{Ts2}, M_{Ts3}, \dots M_{Tst}\}\ \forall t \in \mathbb{N} \qquad 4.\ 10$$

Die Menge aller dem Anwender zur Verfügung stehenden Laserbearbeitungsmaschinen, die für die Laserbearbeitung von dreidimensionalen Werkstücken geeignet sind, kann wie folgt beschrieben werden:

$$M_{Tb} := \{M_{Tb1}, M_{Tb2}, M_{Tb3}, \dots M_{Tbu}\}\ \forall u \in \mathbb{N} \qquad 4.\ 11$$

Und die Menge aller dem Anwender zur Verfügung stehenden Laserbearbeitungsmaschinen, die für die Bearbeitung von Rohrwerkstücken geeignet sind, kann folgendermaßen dargestellt werden:

$$M_{Tt} := \{M_{Tt1}, M_{Tt2}, M_{Tt3}, \dots M_{Ttv}\}\ \forall v \in \mathbb{N} \qquad 4.\ 12$$

Es gilt auch hier, dass die drei Teilmengen in Summe die Menge aller dem Anwender zur Verfügung stehenden Laserbearbeitungsmaschinen bilden. Außerdem ist die Menge von Laserbearbeitungsmaschinen, die sich für eine bestimmte Technologieart eignen, immer eine Teilmenge oder gleich der Menge aller für den Anwender verfügbaren Laserbearbeitungsmaschinen:

$$M := \{M_{Ts} \cup M_{Tb} \cup M_{Tt}\} \forall M_{Ts} \subseteq M und M_{Tb} \subseteq M und M_{Tt} \subseteq M \qquad 4.\ 13$$

Abhängig von den Angaben im Fertigungsauftrag bezüglich der Werkstückform wird eine Technologie gewählt. In diesem Schritt wird bei den drei Knoten die Entscheidungsregel hinsichtlich der Zugehörigkeit einer Laserbearbeitungsmaschine $M_{(i)}$ zu einer der oben erwähnten Teilmengen der Technologiearten überprüft. Bei der Überprüfung werden nur geeignete Laserbearbeitungsmaschinen berücksichtigt, die Teil der entsprechenden Menge $M_{(A)}$ sind, die im ersten Schritt gebildet wurde.

Dabei wird überprüft, ob eine Laserbearbeitungsmaschine für die Fertigung von zwei- bzw. dreidimensionalen Werkstücken oder von Rohrwerkstücken geeignet ist. Die Entscheidungsregeln dienen also der Feststellung, ob eine Laserbearbeitungsmaschine einer der Mengen $\{M_{Ts}\}$, $\{M_{Tb}\}$ oder $\{M_{Tt}\}$ zugehörig ist.

Wenn auf eine der Entscheidungsregeln eine positive Antwort für eine Laserbearbeitungsmaschine folgt, dann wird diese Maschine Teil der Menge der geeigneten Laserbearbeitungsmaschinen $M_{(Apt)}$ und bei den nächsten Knoten weiter betrachtet. Die

formale Beschreibung der Entscheidungsregeln bei den drei Knoten hinsichtlich des Attributs *Technologie* lautet folgendermaßen:

$$if\, M_{(i)} \in M_{Ts}\; dann\; M_{(i)} \in M_{(Apt)}\; oder \qquad 4.\,14$$

$$if\, M_{(i)} \in M_{Tb}\; dann\; M_{(i)} \in M_{(Apt)}\; oder \qquad 4.\,15$$

$$if\, M_{(i)} \in M_{Tt}\; dann\; M_{(i)} \in M_{(Apt)} \qquad 4.\,16$$

$$\forall i \in \mathbb{N}$$

Es gilt auch hierbei, dass die Teilmenge der für eine bestimmte Technologieart geeigneten Laserbearbeitungsmaschinen ein Teil der aus Schritt eins resultierenden Menge aller Laserbearbeitungsmaschinen ist.

$$M_{Apt} \subseteq M_{Ap} \qquad 4.\,17$$

Das Ergebnis des ersten Schritts ist eine Menge von Laserbearbeitungsmaschinen, die für das Laserschneiden geeignet sind $M_{(Ap)}$. Im zweiten Schritt wird zum Beispiel die Technologie für zweidimensionale Werkstücke gewählt. Hier werden also Laserbearbeitungsmaschinen gesucht, die für die Bearbeitung zweidimensionaler Werkstücke $\{M_{Ts}\}$ geeignet sind. Dazu wird der obere Pfad vom zweiten Knoten bis zum ersten Blätterknoten der Technologie verfolgt und die erste Entscheidungsregel 4.15 wird überprüft. Die Laserbearbeitungsmaschinen, bei denen diese Überprüfung die Antwort *Ja* ergibt, werden in die Menge $M_{(Apt)}$ eingefügt und bei den nächsten Knoten weiter betrachtet. Alle Laserbearbeitungsmaschinen, bei denen die Prüfung die Antwort *Nein* ergab, werden bei den nächsten Knoten nicht mehr betrachtet:

$$if\, M_{(i)} \notin M_{Ts}\, dann M_{(i)} \notin M_{(Apt)} \qquad 4.\,18$$

Bei den weiteren Knoten für die anderen Technologiearten werden die Entscheidungsregeln nicht überprüft. Das bedeutet, es wird in diesem Beispiel nur der oberste Ast des Entscheidungsbaumes für die nächsten Schritte weiterverfolgt (siehe Abbildung 4-6).

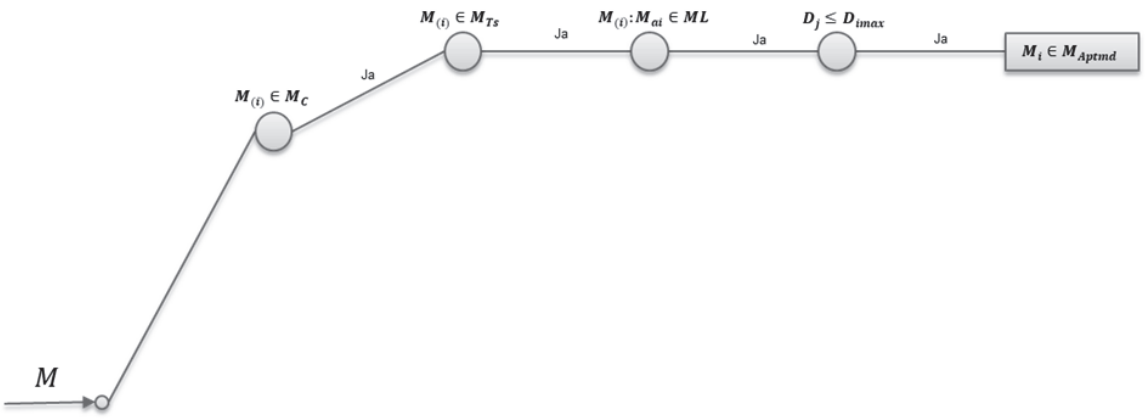

Abbildung 4-6: Entscheidungsbaum zur Maschinenauswahl für Laserschneidprozesse und 2-D-Technologie (eigene Darstellung)

Im dritten Schritt wird das Attribut *Materialart* betrachtet. Hierbei wird überprüft, welche Laserbearbeitungsmaschinen die im Auftrag angegebene Materialart bearbeiten können. Für jede Laserbearbeitungsmaschine ist eine Materialiste (ML) vorhanden, auf der alle bearbeitbaren Materialien vermerkt sind. Hier die formale Beschreibung eines Beispiels einer solchen Materialliste:

$$ML = \{St, Cu, Al, Ag, Sn, \ldots.\} \qquad 4.\ 19$$

Im dritten Schritt wird überprüft, ob sich die im Auftrag M_a angegebene Materialart in der Materialliste (ML) der Laserbearbeitungsmaschine wiederfindet. Wenn diese Entscheidungsregel die Antwort *Ja* ergibt, dann wird diese Maschine Teil der Menge $M_{(Aptm)}$ aller geeigneten Laserbearbeitungsmaschinen. Die formale Beschreibung der Entscheidungsregeln bei diesem Knoten lautet folgendermaßen:

$$F\ddot{u}r M_{(i)}: if M_{ai} \in ML_{(i)}\ dann\ M_{(i)} \in M_{(Aptm)} \qquad 4.\ 20$$

Wenn die Antwort auf diese Frage *nein* lautet, dann wird diese Laserbearbeitungsmaschine in der Folge nicht mehr betrachtet.

$$F\ddot{u}r M_{(i)}: if M_{ai} \notin ML_{(i)} dann M_{(i)} \notin M_{(Aptm)} \qquad 4.\ 21$$

Es gilt hierbei, dass die Teilmengen der für den Umgang mit einer bestimmten Materialart geeigneten Laserbearbeitungsmaschinen ein Teil der Menge aller Laserbearbeitungsmaschinen aus dem zweiten Schritt sind.

$$M_{Aptm} \subseteq M_{Apt} \qquad 4.\ 22$$

Im vierten Schritt wird das Attribut *Materialdicke* betrachtet. Beim Laserschneiden ist die Materialdicke des Werkstücks von zentraler Bedeutung. Dagegen spielt die Materialdicke beim Schweißen und Markieren keine Rolle. Der vierte Schritt in der Maschinenauswahl betrifft demzufolge nur die Laserschneidmaschinen. Das bedeutet, dass der Entscheidungsbaum für das Laserschweißen und -gravieren nach dem dritten Schritt endet. Die Menge der geeigneten Laserbearbeitungsmaschinen M_{Aptm} wird dann als das Ergebnis der Maschinenauswahl bezeichnet.

Bei der weitergehenden Überprüfung der Entscheidungsregeln werden die Laserschneidmaschinen aus der Menge M_{Aptm} berücksichtigt. Hierbei wird überprüft, welche dieser Laserbearbeitungsmaschinen ein Werkstück mit der im Auftrag erwähnten Materialdicke bearbeiten kann. In der Regel wird für jede Laserbearbeitungsmaschine eine maximal bearbeitbare Materialdicke D_{imax} bezogen auf die Materialart festgelegt. Die formale Beschreibung der Entscheidungsregel bei diesem Knoten lautet folgendermaßen:

$$F\ddot{u}r M_{(i)}: if\, D_j \leq D_{imax}\, dann\, M_{(i)} \in M_{(Aptmd)}\ \forall j \in \mathbb{N} \qquad 4.\ 23$$

Wenn diese Entscheidungsregel für eine Laserbearbeitungsmaschine die Antwort *Nein* ergibt, dann wird diese Maschine nicht Teil der Menge der geeigneten Laserbearbeitungsmaschinen $M_{(Aptrd)}$.

$$F\ddot{u}r M_{(i)}: if\, D_j > D_{imax} dann M_{(i)} \notin M_{(Aptmd)} \qquad 4.\ 24$$

Der Entscheidungsbaum wird rekursiv von der Wurzel zu den Blättern generiert. Das Ergebnis ist eine Menge von für das Laserschneiden geeigneten Laserbearbeitungsmaschinen $M_{(Aptmd)}$ sowie eine Menge von Laserschweiß- und –markiermaschinen $M_{(Aptm)}$.

$$M_{(Aptmd)} := \{M_{m1}, M_{m2}, M_{m3}, \dots M_{mx}\} \forall x \in \mathbb{N} und M_{(i)} \in M_c und M_{(i)} \in M_{Ts} und M_{ai} \in ML_{(i)} und D_j \leq D_{imax} \qquad 4.\ 25$$

Das oben erwähnte Beispiel wird nun bis zum dritten und vierten Schritt weiterverfolgt. Angenommen wird, dass der Auftrag einen Laserschneideprozess für zweidimensionale Werkstücke aus Nickel bei einer Materialdicken von 5 mm umfasst. Die Menge der geeigneten Laserbearbeitungsmaschine $M_{(Aptmd)}$ wird dann wie folgt abgebildet:

$$M_{(Aptmd)} := \{M_{m1}, M_{m2}, M_{m3}, \dots M_{mx}\}\ \forall x \in \mathbb{N}\ und\ M_{(i)} \in M_c und\ M_{(i)} \in M_{Ts} und\ M_{ai} = Ni \in ML_{(i)} und\ D_j = 5mm \leq D_{imax} = 10mm \qquad 4.\ 26$$

Zum besseren Verständnis der Maschinenauswahl wird ein weiteres Beispiel angeführt: Der Anwender bekommt einen Laserbearbeitungsauftrag für den Prozess *Laserschweißen*. Es soll ein Rohrwerkstück (*Technologieart*) aus Aluminium (*Materialart*) entstehen. In diesem Fall wird die Menge der geeigneten Laserbearbeitungsmaschinen $M_{(Aptm)}$ wie folgt abgebildet:

$$M_{(Aptm)} := \{M_{m1}, M_{m2}, M_{m3}, \dots M_{mx}\}\ \forall x \in \mathbb{N}\ und\ M_{(i)} \in M_w\ und\ M_{(i)} \in M_{Tt}\ und\ M_{ai} = Al \in ML_{(i)} \qquad 4.\ 27$$

Der Entscheidungsprozess bezüglich der Frage, welche Laserbearbeitungsmaschine in diesem Fall geeignet ist, wird durch den Pfad, der auf der Abbildung 4-7 dargestellt wird, illustriert.

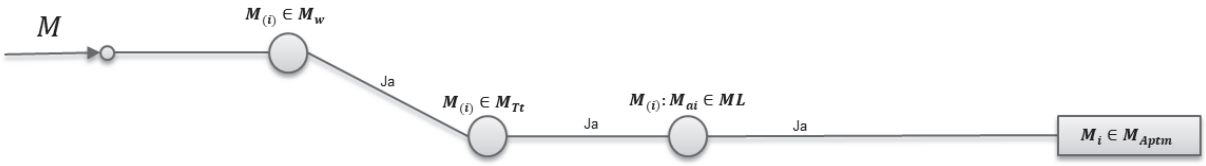

Abbildung 4-7: Entscheidungsbaum zur Maschinenauswahl für Laserschweißprozesse zur Verarbeitung von Rohrwerkstücken aus Aluminium (eigene Darstellung)

Diese Menge von geeigneten Laserbearbeitungsmaschinen $M_{(Aptmd)}$ bzw. $M_{(Aptm)}$ ist schließlich für die Auftragsplanung von Bedeutung.

4.4.2 Lizenzmodellauswahl

Die Technologiedaten liegen in digitaler Form vor, und sie sind Eigentum des Unternehmens, das sie generiert hat. Auf den Technologiedatenmarktplätzen werden mittels unterschiedlicher Lizenzierungsmöglichkeiten Nutzungsrechte für diese Daten angeboten. Dank verschiedener Nutzungsmodelle kann die Art des Bezuges der Technologiedaten den individuellen

Bedürfnissen des Anwenders angepasst werden. Die verschiedenen Lizenzmodelle ermöglichen dem Anwender flexible und optimierte Laserfertigungsprozesse. Die Lizenz ist ein digitaler Vertrag zwischen dem Anwender und dem Technologiedatenmarktplatz, in dem die Bedingungen der Nutzung der Technologiedaten festgeschrieben werden.

Lizenzen schränken ein, was Anwender mit den erworbenen Technologiedaten tun dürfen, und sie begrenzen die Haftung des Anbieters. Eine Lizenzdatei enthält Berechtigungsrichtlinien, auf die sich Lizenznehmer und Lizenzgeber geeinigt haben. Sie beinhaltet zudem neben den Schlüsseln, die zur Realisierung der Nutzungsbedingungen verwendet werden, weitere Identifizierungsinformationen wie zum Beispiel die Kennung des Hardwaresystems oder des Anwenders. Die Schlüssel werden benötigt, um die verschlüsselten Technologiedaten verwenden zu können. Eine Übersicht über die existierenden Lizenzmodelle für digitale Produkte erfolgt in Kapitel 2.4.5. Die untenstehende Übersicht über die verschiedenen Lizenzmodelle (siehe Abbildung 2-12) dient als Ausgangspunkt für die Bestimmung der geeigneten Lizenzmodelle für das elektronische Erwerben von Technologiedaten in der Laserbearbeitung. Das Merkmal Lizenzklasse bezieht sich auf eine Software mit verschiedenen Versionen und regelmäßigen Updates. Bei Technologiedaten handelt es sich aber nicht um klassische Softwareprogramme. Es gibt weder verschiedene Versionen davon noch gibt es Updates. Deshalb ist es auch nicht sinnvoll, bei der Bestimmung der Lizenzmodelle für Technologiedaten von Lizenzklassen zu sprechen. Für jede Technologiedatendatei wird grundsätzlich eine Lizenz über ein Lizenzmodell erworben. Eine Lizenz kann für ein bestimmtes System als Einzelplatz oder für mehrere Systeme als Mehrplatz erstellt werden. Es handelt sich hier ausschließlich um ein Produkt, nämlich die Technologiedaten. Daher wird das Merkmal Pakete ausgeschlossen. Bei Gruppierungslizenzen dürfen mehrere Laserbearbeitungsmaschinen desselben Herstellers und Modells oder mehrere Mitarbeiter die Lizenz gemeinsam nutzen, was dann als eine einzige Nutzung gewertet wird. Das kann zur unklaren Verfolgung der Datennutzung führen. Die Nutzung an mehreren Laserbearbeitungsmaschinen desselben Herstellers und Modells wird mit der Eigenschaft Mehrplatz abgedeckt. Im Konzept dieser Dissertation bekommt jede Laserbearbeitungsmaschine eine individuelle Identifikationsnummer, die in hardwaregebundenen Lizenzen einzugeben ist. Aufgrund dessen wird dieses Merkmal ebenfalls ausgeschlossen. Bei den Attributen Lizenztyp und Lizenzmetrik entsprechen die Lizenzmerkmale jenen von Softwareprogrammen (siehe Kapitel 2.4.5).

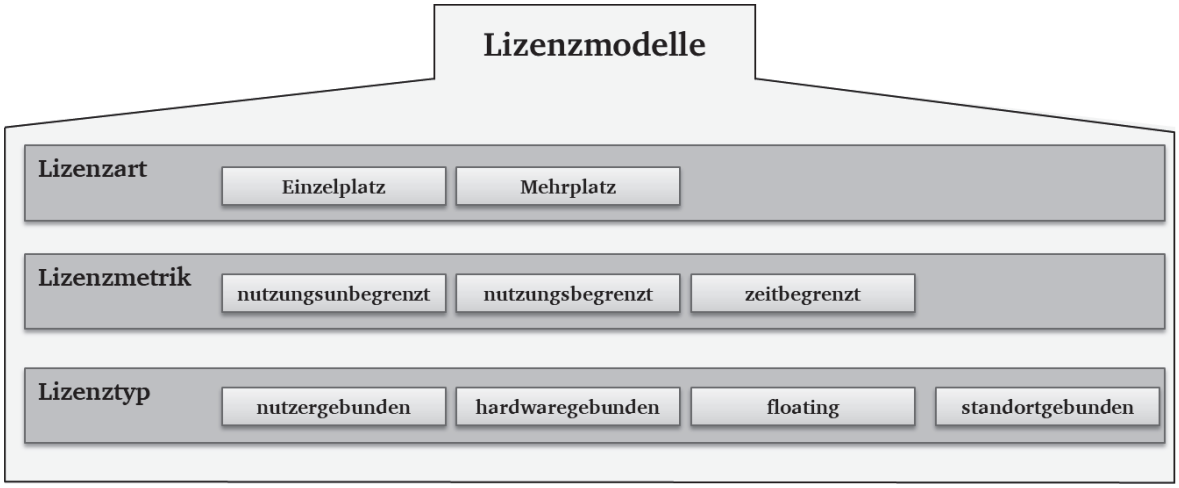

Abbildung 4-8: Lizenzmodelle für den Technologiedatenerwerb (eigene Darstellung)

Für die Abbildung 4-8 wurden Attribute und Merkmale von für den Erwerb von Technologiedaten geeigneten Lizenzmodellen zusammengestellt. Diese Abbildung dient als Basis für die Konzipierung der Lizenzmodellauswahl im nächsten Unterkapitel. Ein Assistenzsystem wird entwickelt, um dem Anwender Handlungsempfehlungen zur Auswahl der optimalen Lizenzmodelle bereitzustellen. Schließlich werden abhängig von den Auftragsinformationen und den Erfahrungen des Anwenders unterschiedliche Lizenzmodelle empfohlen. Zu den Attributen *Lizenzart, Lizenztyp* und *Lizenzmetrik* werden auf Basis von Informationen über den Fertigungsauftrag die entsprechenden Werte zugewiesen.

Die empfohlenen Lizenzmodelle werden durch die Festlegung unterschiedlicher Merkmalwerte für die drei Attribute *Lizenzart, Lizenztyp* und *Lizenzmetrik* auf Grundlage der Antworten des Anwenders gebildet. In Bezug auf das Attribut *Lizenzart* kann zwischen den Merkmalwerten Einzelplatz und Mehrplatz ausgewählt werden. Das Assistenzsystem kann in diesem Fall auf der Grundlage des angegebenen Auslieferungsdatums zwischen Lizenzen wählen, die ausschließlich für eine bestimmte Laserbearbeitungsmaschine gültig sind, und solchen, die für mehrere Laserbearbeitungsmaschinen verwendet werden können. Die Auftragsinformationen, insbesondere das Auslieferungsdatum, sind entscheidend für die Bestimmung dieses Attributes. Wenn beispielsweise ein Auftrag dringend und zeitnah ausgeliefert werden muss, ist es ratsam, ihn auf mehreren Laserbearbeitungsmaschinen zu bearbeiten als der Lizenzart als Mehrplatz zu wählen. Der *Lizenztyp* beschreibt die Art und Weise, wie eine Lizenz an eine Maschine, einen Nutzer, ein Netzwerk oder einen Standort gebunden wird. Die genaue Bindung hängt von der Flexibilität der Auftragsbedingungen ab.

Das Attribut *Lizenzmetrik* betrifft die Nutzungsbedingungen der Technologiedaten. Die Auftragsinformationen sind von zentraler Bedeutung für die Bestimmung dieses Attributs. Abhängig vom Auftragsvolumen und der Auftragshäufigkeit wird zwischen nutzungsbegrenzten, nutzungsunbegrenzten und zeitbegrenzten Lizenzmetriken unterschieden. Wenn zum Beispiel ein Auftrag über eine höhere Anzahl von Werkstücken vorliegt – das Auftragsvolumen also groß ist –, kann der Anwender unbegrenzt gültige Lizenzen auswählen, um so die Lizenzkosten pro Stück zu minimieren. Wenn hingegen ein Auftrag über eine kleinere Stückzahl vorliegt, ist es effektiver, nutzungsbegrenzte Lizenzen zu erwerben. Die Häufigkeit des Auftrags spielt auch bei der Wahl zwischen zeit- und nutzungsbegrenzten Lizenzen eine wichtige Rolle. Wenn ein Auftrag innerhalb eines bestimmten Zeitraums häufig erteilt wird, dann wird das Merkmal „zeitbegrenzt“ für die Lizenzmetrik bevorzugt.

Entwicklung des wissensbasierten Assistenzsystems

Nach Abschluss des Systemelementes Maschinenauswahl wird die Menge der geeigneten Laserbearbeitungsmaschinen $M_{(Aptmd)}$ abhängig von dem Prozessart $M_{(Aptm)}$ bestimmt. Für jede Laserbearbeitungsmaschine in dieser Menge muss nun geprüft werden, ob die benötigten Technologiedaten vorhanden sind – und wenn ja, ob für die vorhandenen Technologiedaten eine gültige Lizenz erworben wurde. Für den Fall, dass keine Technologiedaten vorhanden sind oder dass die existierenden Lizenzen nicht gültig sind, muss nach einem passenden Lizenzmodell gesucht werden. Diese Vorgänge werden im Folgenden beispielhaft dargelegt. Das Ergebnis dieses Prozesses wird anschließend als Handlungsempfehlung hinsichtlich des Erwerbs von Lizenzmodellen zur Verfügung gestellt.

Für die Gestaltung des Auswahlverfahrens wird ein wissensbasiertes Assistenzsystem konzipiert. Dieses System wird bei der Ermittlung passender Lizenzmodelle angewendet, indem es die Charakteristika und Anforderungen des Auftrags analysiert und das passendste Lizenzmodell empfiehlt. Dazu werden die Auswahlkriterien eingehend untersucht und festgelegt. Die Bestimmung der geeigneten Lizenzmodelle erfolgt auf Basis der Kenntnisse über die verfügbaren Lizenzmodelle und über die festgelegten Auswahlkriterien und -regeln. Diese Informationen sind Teil der Wissensbasis des Assistenzsystems. Die Inferenzmaschine verwendet dieses gespeicherte Wissen zur Bewertung der Auftragseigenschaften, wie etwa Liefertermin, Auftragsvolumen und Anforderungen an dessen Flexibilität, um das für den Auftrag am besten geeignete Lizenzmodell zu empfehlen. Im Folgenden werden die Auswahlkriterien detailliert dargelegt:

1. Die Zeitkritikalität des Auftrags: Die Zeitkritikalität eines Auftrags lässt sich durch den Vergleich des vorgegebenen Liefertermins mit dem gegenwärtigen Datum ermitteln. Diese Information wird zur Definition des ersten Attributs des Lizenzmodells, der spezifischen Lizenzart, herangezogen. Bei einer hohen Zeitkritikalität eines Auftrags kann die Aufteilung der Laserbearbeitung auf mehrere Laserbearbeitungsmaschinen empfohlen werden, um die Durchführung des Auftrags zu beschleunigen. Ist der Auftrag hingegen nicht von hoher Dringlichkeit, kann er auf einer einzelnen Laserbearbeitungsmaschine durchgeführt werden. Es ist die Verantwortung eines versierten Anwenders, auf Grundlage einer vorab über die Benutzeroberfläche festgelegten Lieferfrist zu beurteilen, ob ein Auftrag als zeitkritisch betrachtet wird. Beispielsweise könnte der Anwender eine Lieferfrist von einem Monat als Indikator für eine hohe Zeitkritikalität interpretieren und entsprechend in den Systemeinstellungen festlegen. Dieser Indikator kann jedoch zu jedem Zeitpunkt vom Benutzer angepasst werden. Das Assistenzsystem wird, unter Berücksichtigung des eingegebenen Liefertermins sowie der in den Systemeinstellungen definierten Lieferfrist, ermitteln, ob den eine zeitkritische Abwicklung des Auftrags notwendig ist.

2. Das Volumen des Auftrags: Das zweite Attribut eines Lizenzmodells, die Lizenzmetrik, kann basierend auf dem Volumen des Auftrags definiert werden. In diesem Kontext wird der Benutzer aufgefordert, einzuschätzen, ob das Auftragsvolumen als groß oder klein zu bewerten ist. Wenn das Auftragsvolumen als klein eingestuft wird, empfiehlt das System eine nutzungsbegrenzte Lizenz. Dies impliziert, dass die Technologiedaten nur für eine beschränkte Anzahl von Anwendungen genutzt werden dürfen. Wenn das Auftragsvolumen hingegen als groß eingestuft wird, wird entweder eine nutzungsunbegrenzte oder eine zeitlich begrenzte Lizenz empfohlen. Die Festlegung der Grenze zur Kennzeichnung eines Auftrags als klein oder groß liegt im Ermessen eines erfahrenen Anwenders und kann individuell über die Systemeinstellungen vorgenommen werden. Diese Einstellung kann flexibel angepasst werden, um den aktuellen Markt- und Fabrikbedingungen gerecht zu werden.

3. Die Häufigkeit des Auftrags: Wenn bekannt ist, wie oft ein Auftrag beim Anwender eingeht, kann das Assistenzsystem präzisere Empfehlungen zur Lizenzmetrik abgeben.

    a. Wenn Aufträge nur einmalig oder selten eingehen, wird ein nutzungsbegrenzter Lizenztyp empfohlen. Die Technologiedaten sind beim Kauf einer solchen Lizenz nur für eine begrenzte Anzahl von Nutzungen vorgesehen.

b. Wenn die Aufträge oft eingehen, dann werden nutzungs- oder zeitbegrenzte Lizenzen empfohlen. Die Regelmäßigkeit der Auftragseingänge innerhalb eines festgelegten Zeitraums (z. B. eines Jahres) wird zur feineren Bestimmung der Lizenzmetrik herangezogen. Wenn der Auftrag jährlich eingeht, wird eine Lizenzmetrik ohne zeitliche Einschränkungen empfohlen, also eine nutzungsunbegrenzte. Dadurch können die Technologiedaten uneingeschränkt und ohne zeitliche Begrenzung genutzt werden. Wenn dagegen der Auftrag unregelmäßig kommt, dann wird eine zeitbegrenzte Lizenz empfohlen. Die Technologiedaten dürfen dann nur innerhalb eines begrenzten Zeitraums genutzt werden, in diesem jedoch beliebig oft.

4. Die Flexibilität des Auftrags: Die Bestimmung des Lizenztyps sollte auf der Grundlage der Beantwortung der Frage, ob es Einschränkungen in Bezug auf den Anwender, die Laserbearbeitungsmaschine, die Fabrik oder den Standort gibt, erfolgen. In diesem Zusammenhang gibt es fünf Auswahlmöglichkeiten:

   a. Wenn der Auftrag von einem bestimmten Mitarbeiter bearbeitet werden muss, wird der empfohlene Lizenztyp als nutzergebunden festgelegt. Das bedeutet, dass die Technologiedaten nur von diesem bestimmten Mitarbeiter genutzt werden dürfen, der berechtigt ist, den Auftrag zu bearbeiten.

   b. Wenn der Auftrag auf einer bestimmten Laserbearbeitungsmaschine bearbeitet werden muss, wird der empfohlene Lizenztyp als hardwaregebunden festgelegt. Dies besagt, dass die Technologiedaten nur auf der spezifischen Laserbearbeitungsmaschine verwendet werden dürfen, die zur Bearbeitung des Auftrags vorgesehen ist.

   c. Wenn der Auftrag in einem bestimmten Fabrik oder Werk bearbeitet werden muss, wird der empfohlene Lizenztyp als Floating festgelegt. Dies bedeutet, dass die Technologiedaten von jedem Anwender und auf jeder Maschine innerhalb des spezifischen Werks verwendet werden dürfen, das für die Bearbeitung des Auftrags vorgegeben ist.

   d. Wenn der Auftrag an einem bestimmten Standort bearbeitet werden muss, empfiehlt sich der Lizenztyp standortgebunden. Dies bedeutet, dass die Technologiedaten ausschließlich an dem Ort verwendet werden dürfen, der für die Bearbeitung des Auftrags vorgesehen ist.

e. Falls keine Beschränkungen vorliegen, werden alle Attribute des Lizenztyps berücksichtigt. Das Assistenzsystem sollte dem Anwender daher alle 4 genannten Optionen des Lizenztyps in seiner Empfehlung präsentieren.

Basierend auf den zuvor festgelegten Auswahlkriterien wurden Fragen formuliert, die dem Anwender über die Benutzerschnittstelle gestellt werden sollten, um ihn bei der Auswahl des geeigneten Lizenzmodells zu unterstützen. Der Entscheidungsbaum des Assistenzsystems, der in Abbildung 4-9 dargestellt ist, umfasst die definierten Fragen, mögliche Antworten und die empfohlenen Lizenzmodelle.

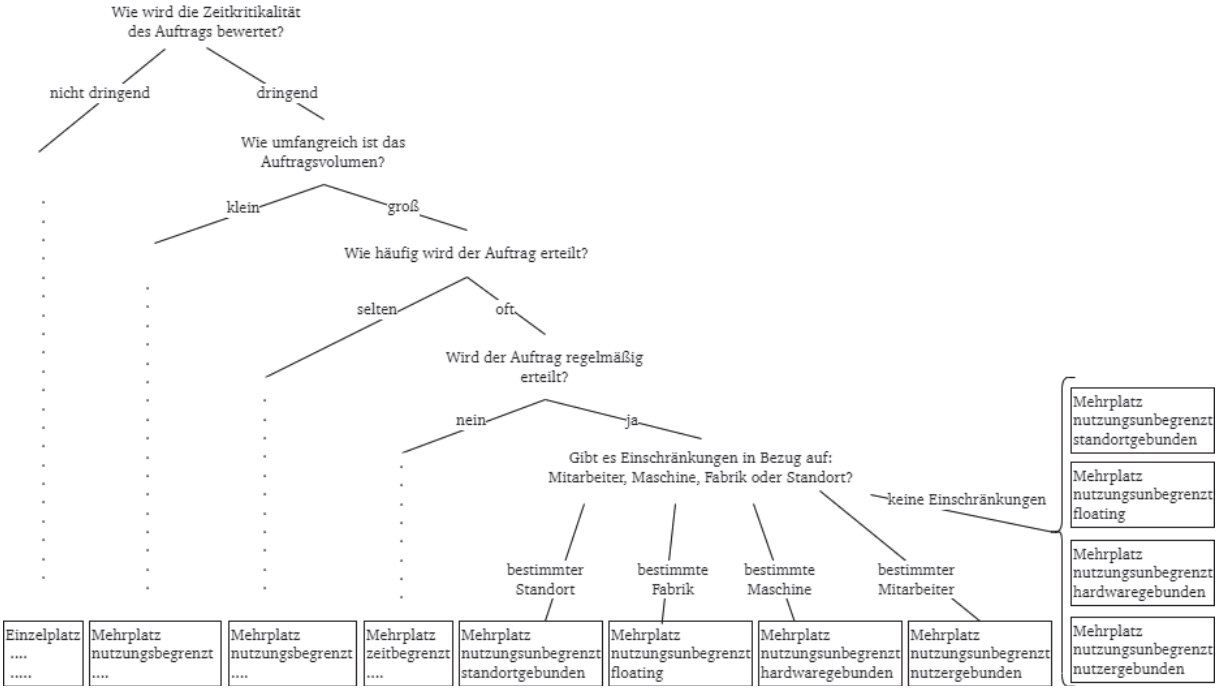

Abbildung 4-9: Entscheidungsbaum des Assistenzsystems (eigene Darstellung)

Wie bereits erwähnt wurde, wird das Lizenzmodell von Merkmalwerten der drei Attribute *Lizenzart*, *Lizenztyp* und *Lizenzmetrik* strukturiert. Das heißt, um ein Lizenzmodell zu bestimmen, muss einer der dargestellten Merkmalwerte der jeweiligen Attribute gewählt werden. Für das Attribut *Lizenzart* werden zwei Merkmalwerte definiert, für *Lizenztyp* sind es vier und für *Lizenzmetrik* drei. Die maximale Größe der Menge aller möglichen Optionen für Lizenzmodelle wird mittels folgender Formel berechnet:

$$l_{max} = 2 * 4 * 3 = 24 \qquad 4.\ 28$$

Diese Menge enthält im vorliegenden Beispiel 24 verschiedene Modelle. Das ergibt sich aus folgender Formel, die mithilfe der logischen Operationen „UND“ (auch Konjunktion genannt) und „ODER“ (auch als Disjunktion bezeichnet) dargestellt wird:

$$l_{max} = \{E \vee M\} \wedge \{A \vee H \vee F \vee S\} \wedge \{U \vee N \vee Z\} =$$
$$\begin{Bmatrix} L_1(EAU), L_2(EAN), L_3(EAZ), L_4(MAU), L_5(MAN), L_6(MAZ), L_7(EHU), L_8(EHN), L_9(EHZ), \\ L_{10}(MHU), L_{11}(MHN), L_{12}(MHZ), L_{13}(EFU), L_{14}(EFN), L_{15}(EFZ), L_{16}(MFU), L_{17}(MFN), \\ L_{18}(MFZ), L_{19}(ESU), L_{20}(ESN), L_{21}(ESZ), L_{22}(MSU), L_{23}(MSN), L_{24}(MSZ) \end{Bmatrix}$$

4. 29

Nun folgt ein Beispiel für die Bildung einer Menge von empfohlenen Lizenzmodellen. Wird zum Beispiel ein Auftrag als nicht dringend bezeichnet, wird der Einzelplatz als Merkmalwert des Attributes *Lizenzart* gewählt. In Bezug auf die Flexibilität werden die Hardwaregebundenheit oder Floating als Merkmalwert des Attributes *Lizenztyp* festgelegt. Das Auftragsvolumen wird als klein bezeichnet, und deshalb wird eine begrenzte Nutzung als Merkmalwert bei *Lizenzmetrik* gewählt. Die Größe der Menge der möglichen Lizenzmodelle wird anschließend durch die Formel 4. 31 berechnet:

$$l = 1 * 2 * 1 = 2 \qquad 4.30$$

Nach der Konzipierung des zweiten Systemelementes *Lizenzauswahl* folgt nun das Systemelement *Datensicherheit*.

4.4.3 Datensicherheit

Die steigende Zahl der vernetzten Systeme in der Industrie führt dazu, dass immer mehr Anforderungen an die IT-Sicherheit gestellt werden. Ausgehend davon müssen die IT-Sicherheitskonzepte frühzeitig im Entwicklungsprozess berücksichtigt werden [5]. In diesem Unterkapitel wird das Systemelement „Datensicherheit“ aufbauend auf den im Kapitel 2.4 bereits vorgestellten Sicherheitsmaßnahmen konzipiert. Der Vorgang der Lizenzierung ist ein interessantes und anspruchsvolles Thema. Im Folgenden steht demnach die Lizenzierung von erworbenen Technologiedaten im Fokus.

Mit einer Lizenz wird in der Regel nicht ein digitales Produkt erworben, sondern bloß das Recht, es zu nutzen. Die elektronische Vermarktung und Verbreitung von Technologiedaten über

Technologiedatenmarktplätze erfolgt mittels solcher Lizenzen. Gemäß Fachliteratur existiert kein einheitliches Standardformat für Lizenzen. Eine Lizenzvereinbarung ist im Grunde ein Dokument, das rechtsverbindliche Richtlinien für die Nutzung von digitalen Produkten enthält. Die Lizenzverträge beinhalten demgemäß Bestimmungen zu den Produktherstellern, zu den Nutzungsrechten, zur Gewährleistung, zum Urheberrecht, zur Haftung, zur legitimen Art und Weise der Verwendung der Daten und zu den Folgen einer missbräuchlichen Nutzung. Im weiteren Verlauf gerät die konzeptionelle Verwendung der Nutzungsrechte von Technologiedaten durch die Anwender in den Blick. Die Lizenzen schränken die Nutzung der erworbenen Technologiedaten ein, sie spezifizieren die Zugriffsrechte der Anwender. So geben die Lizenzgeber „die Technologiedatenmarktplätze“ die Nutzungszeiträume und/oder -bedingungen vor.

Das Lizenzierungskonzept zum Schutz der erworbenen Technologiedaten baut auf den Grundlagen der Lizenzierung von Softwareprodukten auf. Der Fokus liegt im Folgenden auf der technischen Umsetzung der zwischen dem Technologiedatenmarktplatz und dem Anwender vereinbarten Nutzungsbedingungen. Diese Nutzungsbedingungen unterscheiden sich je nach Lizenzmodell. Wie bereits im Kapitel 2.4.13 erwähnt wurde, wird für die Entwicklung des Lizenzierungskonzeptes von Technologiedaten in dieser Dissertation auf softwarebasierte Ansätze zurückgegriffen, basierend auf Blockchain- und Smart-Contract-Technologien. Das entwickelte Konzept zur Datensicherheit wird in der Abbildung 4-10 dargestellt. Nun wird das Lizenzierungskonzept zur sicheren Nutzung der erworbenen Technologiedaten durch den Anwender vorgestellt. Die richtigen Systeme zum Auslesen und zum Auswerten einer Lizenz müssen dafür bereitstellt werden. Es handelt sich dabei um den Lizenz-Vertrauensagenten, den Lizenzmanager und das Kryptosystem – diese drei Entitäten werden im Folgenden beleuchtet.

Das entwickelte Assistenzsystem liefert dem Anwender in der Planungsphase der Laserfertigung Handlungsempfehlungen bezüglich der Auswahl einer geeigneten Laserbearbeitungsmaschine und eines Lizenzmodells. Die Konzipierung der Systemelemente „Maschinenauswahl“ und „Lizenzmodellauswahl“ erfolgte in den vorherigen Kapiteln 4.4.1 und 4.4.2. Basierend auf diesen Handlungsempfehlungen werden Technologiedaten für bestimmte Laserbearbeitungsmaschinen mit einem bestimmten Lizenzmodell erworben. Der Erwerb von Technologiedaten oder von neuen Lizenzen geht mittels des ERP (engl. Enterprise Resource Planning) vonstatten.

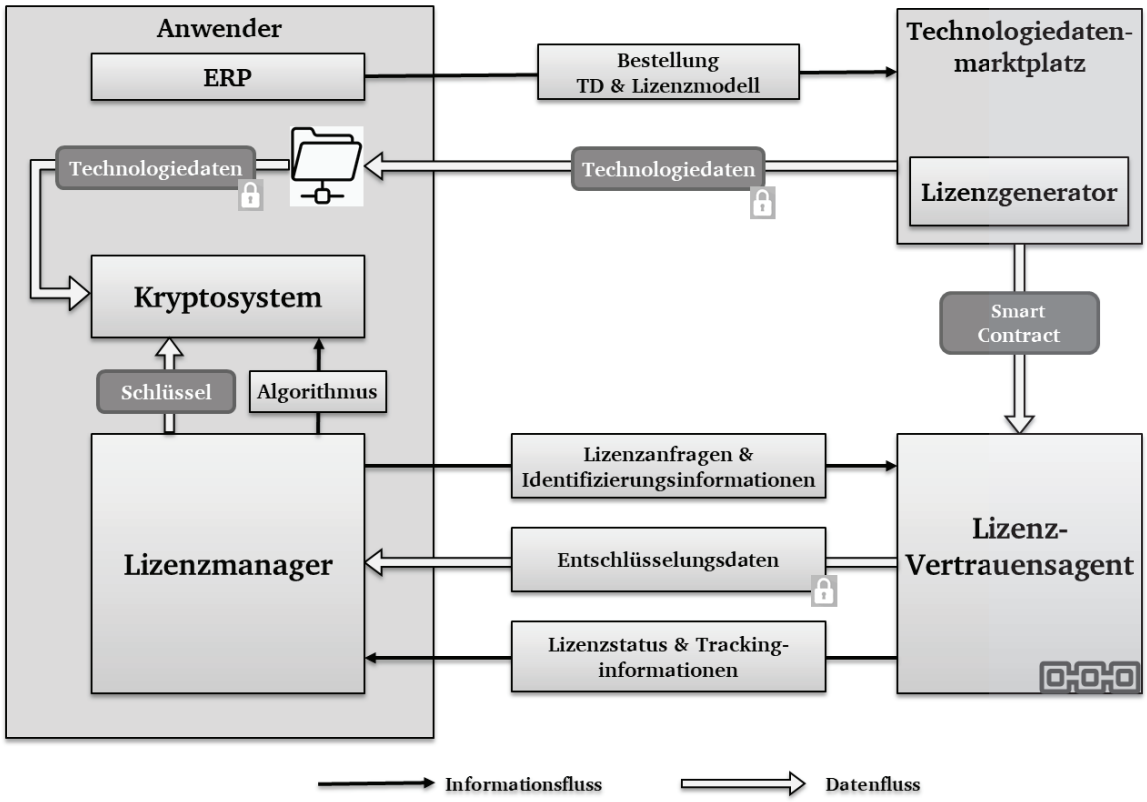

Abbildung 4-10: Lizenzierungskonzept (eigene Darstellung)

Beim erfolgreichen Erwerb werden die Technologiedaten verschlüsselt, vorbereitet und anschließend an den Anwender gesendet. Gleichzeitig werden Daten über die Bestellung, den Anwender (Käufer), das Lizenzmodell und die gekauften Technologiedaten gespeichert. Ab diesem Moment wird ein digitaler intelligenter Vertrag (engl. Smart Contract) generiert und dem Anwender auf dem Lizenz-Vertrauensagenten zur Verfügung gestellt. Der Lizenz-Vertrauensagent verwaltet und speichert die Lizenzen von der Bereitstellung bis zur Nutzung im Blockchain-Netzwerk. Außerdem wird ein individuelles Lizenztoken generiert und dem Anwender zugewiesen. Ein Lizenztoken ist ein eindeutiger Identifikator, der anhand verschiedener Faktoren wie zum Beispiel Kunden-ID, Bestellungs-ID und/oder Maschinen-ID generiert wird. Für die Generierung von Lizenzen und die Bereitstellung des intelligenten Vertrags auf dem Lizenz-Vertrauensagenten wird ein Lizenzgenerator verwendet.

Die Daten werden benötigt, damit die Produktion anlaufen kann. In der Lizenz sind verschlüsselte Informationen abgelegt, die den Anwender in die Lage versetzen, die von ihm erworbenen Technologiedaten unter den im Lizenzmodell festgelegten Bedingungen zu nutzen.

Lizenzen können über ein Webinterface direkt vom Anwender beim Bedarf sicher abgerufen werden. Beim Abrufen von Lizenzen kommuniziert der Lizenzmanager mit dem Lizenz-Vertrauensagenten. Dabei überprüft der Lizenz-Vertrauensagent mittels des jeweiligen intelligenten Vertrags, ob der Anwender dazu berechtigt ist. Dafür müssen zunächst Informationen über den Anwender und eventuell über die Laserbearbeitungsmaschine eingegeben werden. Der Anwender loggt sich mit seinen Zugangsdaten ein. Zum Beispiel gibt der Anwender seine Benutzeridentifikation, sein Passwort und seine Kundennummer ein. Wenn eine Lizenz an eine bestimmte Laserbearbeitungsmaschine gebunden ist, wird der Anwender gebeten, Informationen über diese Laserbearbeitungsmaschine, also die Systemkennung, einzugeben. Bei erfolgreicher Identifizierung des Anwenders bzw. der Laserbearbeitungsmaschine wird der bereitgestellte intelligente Vertrag automatisch ausgeführt. Wenn die Lizenz noch gültig ist, gibt der Lizenz-Vertrauensagent die Entschlüsselungsdaten zusammen mit sämtlichen weiteren Informationen zurück, die für die Entschlüsselung der Technologiedaten erforderlich sind. Diese Entschlüsselungsinformationen werden mit dem öffentlichen Schlüssel des Anwenders verschlüsselt und danach versendet.

Der Lizenz-Vertrauensagent führt Protokoll (engl. Log) über den Status der Lizenz, zum Beispiel darüber, wann eine Lizenz bestellt, generiert, versandt, ausgetauscht oder abgerufen wurde. Diese Informationen sind für die Verfolgung und Beurteilung von Lizenzanfragen oder bei Missverständnissen sehr wichtig. Wenn zum Beispiel eine Lizenz an eine bestimmte Laserbearbeitungsmaschine gebunden ist und die Maschine ausfällt, wird dokumentiert, dass die alte Lizenz ungültig geworden ist und dass eine neue Lizenz mit neuen Identifikationsinformationen generiert werden muss. Diese Informationen stehen dem Anwender zur Verfügung, denn der Lizenzmanager stellt ihm eine Benutzerschnittstelle zur Verfolgung der Lizenznutzung bereit. Diese informiert den Anwender über den Status und die Gültigkeit der erworbenen Lizenzen. Beispielsweise kann der Anwender eine Benachrichtigung erhalten, wenn Lizenzen in Kürze ihre Gültigkeit verlieren und erneuert werden müssen.

Nur wenn die Lizenzdateien vom Anwender ausgelesen werden können, können die erworbenen Technologiedaten genutzt werden. Deshalb ist es wichtig, dass sowohl auf dem Technologiedatenmarktplatz als auch beim Anwender die gleichen kryptografischen Methoden zur Anwendung kommen. Vorab bedarf es einer entsprechenden Abstimmung, um die Technologiedaten entschlüsseln zu können. Die Entschlüsselung der Technologiedaten erfolgt mittels eines Kryptosystems, das auf einem Server, auf spezieller Hardware oder direkt auf der Laserbearbeitungsmaschine installiert werden kann. Bei Bedarf greift das Kryptosystem auf einen gemeinsamen Ordner zu und holt die verschlüsselten Technologiedaten ab. Wie schon

erwähnt wird eine Anfrage an den Lizenz-Vertrauensagenten geschickt, um die benötigte Lizenz bzw. die Entschlüsselungsinformationen – die Schlüssel und den kryptografischen Algorithmus – zu erhalten, die für die Nutzung der Technologiedaten erforderlich sind. Bei erfolgreicher Anmeldung und Gültigkeit der Lizenz sendet der Lizenz-Vertrauensagent die angefragten Entschlüsselungsdaten zurück, die an das Kryptosystem weitergeleitet werden, um die Technologiedaten zu entschlüsseln. Das Kryptosystem entschlüsselt die Technologiedaten mit Hilfe der gesendeten Schlüssel und des kryptografischen Algorithmus. Schließlich werden die entschlüsselten Technologiedaten als Klartext an die Maschinensteuerung gesendet und der Fertigungsprozess kann beginnen.

Um Technologiedaten vor unerlaubtem Zugriff zu schützen, wird auf End-to-End-Verschlüsselung (engl. End-to-End Encryption, „E2EE“) zurückgegriffen. Hierbei erfolgen die Verschlüsselung auf dem Technologiedatenmarktplatz und die Entschlüsselung erst unmittelbar vor der Verwendung. Das Lizenzierungskonzept wurde entwickelt, um einen sicheren Umgang mit den Technologiedaten zu ermöglichen. Um dies zu erreichen, verwendet das System eine Vielzahl von kryptographischen Algorithmen, Schlüsseln und digitalen Signaturen. Mit Hilfe von Kryptoalgorithmen können Informationen so verschlüsselt werden, dass es ohne den entsprechenden Schlüssel praktisch unmöglich ist, sie zu lesen oder zu verändern [154]. Nun werden die kryptografischen Operationen zum Schutz von Technologiedaten aufbauend auf den technischen Schutzmaßnahmen, die in Kapitel 2.4.2 dargelegt wurden, vorgestellt.

Technische Maßnahmen

Die Vertraulichkeit der Daten wird daran bemessen, dass die verschlüsselten Daten nicht von unberechtigten Teilnehmern decodiert werden können. Um dieses Ziel zu erreichen, müssen die richtigen Standards und Verschlüsselungsalgorithmen ausgewählt werden, wobei immer auf die Verwendung von starken Schlüsseln und auf eine sichere Verwaltung dieser Schlüsseln zu achten ist. Der Anwender kauft bloß die Nutzungsrechte für die Technologiedaten. Die Daten werden auf dem Technologiedatenmarktplatz verschlüsselt und dem Anwender geschickt. Die dazugehörige Lizenz wird auf dem Lizenz-Vertrauensagenten als intelligenter Vertrag bereitgestellt. Durch die Verschlüsselung (engl. Encryption) werden die Technologiedaten geschützt. Sie werden auch in verschlüsselter Form gespeichert, womit sie geheim bleiben. Grundsätzlich existieren drei Hauptverschlüsselungsverfahren: das symmetrische, das asymmetrische und das hybride Verfahren. Diese unterscheiden sich hinsichtlich des Schlüsselaufbaus, der Performance und ihrer Zuverlässigkeit.

In dieser Dissertation wird in erster Linie das hybride Verschlüsselungsverfahren angewandt, das sich als sehr effizient und sicher erwiesen hat. Dieses Verfahren basiert auf einer Mischung aus asymmetrischen und symmetrischen Schlüsseln zur Ver- und Entschlüsselung von Technologiedaten. Asymmetrische Schlüssel sind dabei für die Verschlüsselung des symmetrischen Schlüssels zuständig, welcher wiederum zur Verschlüsselung der Technologiedaten genutzt wird. Durch dieses Verfahren wird eine zusätzliche Sicherheitsebene eingezogen, da der symmetrische Schlüssel nie unverschlüsselt übertragen wird. Das Verfahren zur Ver- und Entschlüsselung der Technologiedaten wird in der Abbildung 4-11 dargestellt.

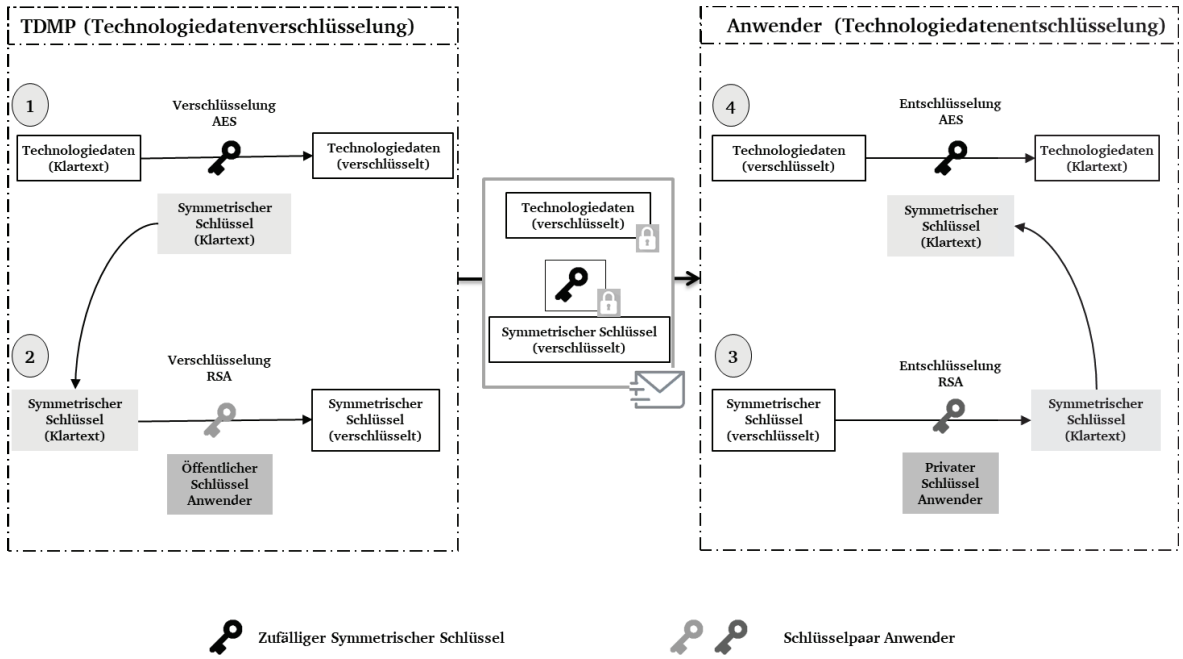

Abbildung 4-11: Ver- und Entschlüsselung der Technologiedaten (eigene Darstellung)

Die Verschlüsselung von Technologiedaten auf dem Technologiedatenmarktplatz erfolgt in zwei Schritten: Zunächst wird ein zufälliger symmetrischer Schlüssel K generiert. Dieser Schlüssel K wird dann in das symmetrische Kryptosystem AES eingesetzt, um die Technologiedaten TD zu verschlüsseln. Hierbei entsteht der Chiffretext c_i. Der Zugriff auf die Technologiedaten ist an den besagten Schlüssel K sowie an den Algorithmus AES gebunden (siehe Formel 4.31).

$$c_i = AES(K; TD) \forall \qquad 4.31$$

c_i: chiffrierte Technologiedaten und K: zufälliger symmetrischer Schlüssel

Im zweiten Schritt wird der symmetrische Schlüssel dann anhand des öffentlichen Schlüssels des Anwenders asymmetrisch verschlüsselt. Hierfür wird der RSA-Algorithmus zur Sicherung der Entschlüsselungsinformationen verwendet:

$$K_c = RSA(e, K) \forall \qquad 4.\ 32$$

K_c: verschlüsselter symetrischer Schlüssel und e: öffentlicher Schlüssel des Anwenders

Die Technologiedaten bleiben innerhalb des Wirkungsbereiches des Anwenders verschlüsselt. Das Kryptosystem darf nur bei einer berechtigten Anfrage auf den Schlüssel der Technologiedaten zugreifen und soll Letztere erst unmittelbar vor ihrer Verwendung an der Laserbearbeitungsmaschine entschlüsseln. Im Folgenden wird das Entschlüsselungsverfahren von Technologiedaten vorgestellt.

Zunächst wird der symmetrische Schlüssel K_c mittels des privaten Schlüssels des Anwenders d im Kryptosystem entschlüsselt. Hierfür wird der Algorithmus *RSA* verwendet.

$$K = RSA^{-1}(d; K_c) = k \ \forall\ d \text{: privater Schlüssel des Anwenders} \qquad 4.\ 33$$

Danach wird der symmetrische Schlüssel K verwendet, um die Technologiedaten c_i lesbar zu machen.

$$TD = AES^{-1}(K; c_i) = TD \qquad 4.\ 34$$

Unmittelbar vor dem Start werden schließlich die Technologiedaten im Klartext an die Steuerung der Laserbearbeitungsmaschine gesendet.

Digitale Signaturen spielen eine entscheidende Rolle bei der Überprüfung der Authentizität und Integrität von Technologiedaten und Entschlüsselungsinformationen. Diese Signaturen werden normalerweise mit dem privaten Schlüssel des Technologiedatenmarktplatzes erstellt und mit dem entsprechenden öffentlichen Schlüssel verifiziert. Dieses Verfahren gewährleistet, dass die Daten nicht verändert oder manipuliert wurden und bestätigt deren Herkunft. Der ECDSA-Algorithmus (*Elliptic Curve Digital Signature Algorithm*) wird aufgrund seiner hohen Sicherheit empfohlen, um digitale Signaturen zu generieren. Dieser asymmetrische Verschlüsselungsalgorithmus weist ähnliche Sicherheitseigenschaften wie RSA auf, nutzt jedoch kleinere Schlüsselgrößen, was ihn besonders effizient in ressourcenbeschränkten Umgebungen

macht. Die Sicherheit von ECDSA basiert auf der Schwierigkeit des Lösens des diskreten Logarithmus in der Theorie der elliptischen Kurven, was ihm eine erhöhte Resistenz gegenüber Angriffen durch Quantencomputer verleiht [155]. Daher ist ECDSA eine weitverbreitete Methode zur Erzeugung digitaler Signaturen in der modernen Kryptographie. Die benötigte Schlüssellänge für ECDSA hängt von der gewünschten Sicherheitsstufe und der spezifischen elliptischen Kurve ab, die zur Erzeugung der Schlüssel verwendet wird.

Die Schlüssel spielen eine zentrale Rolle bei kryptografischen Operationen, da sie für die Verschlüsselung und Entschlüsselung von Daten sowie für die Überprüfung digitaler Signaturen verwendet werden [156]. In dieser Dissertation orientiert sich die gewählte Schlüssellänge an den Empfehlungen des NIST (National Institute of Standards and Technology), die sich auf Algorithmen und Schlüsselgrößen für Schlüsselpaare beziehen, die in der Public Key Infrastructure (PKI) und bei Infrastrukturkomponenten Anwendung finden [157]. Bei der Anwendung von AES wird ein 256-Bit-AES-Schlüssel verwendet. Die Schlüssellänge in RSA-Systemen kann zwischen 1024 und 4096 Bit variieren. Allerdings kann die Nutzung von Schlüsseln mit größerer Länge die Systemleistung negativ beeinflussen. Aus diesem Grund wird meist eine Schlüssellänge von 2048 Bit eingesetzt, um ein ausgewogenes Verhältnis zwischen Sicherheit und Leistung zu gewährleisten. Im Kontext von ECDSA wird in der Regel ein 256-Bit-ECDSA-Schlüssel verwendet. Dieser bietet ein Sicherheitsniveau, das vergleichbar ist mit einem 3072-Bit-RSA-Schlüssel [155].

Um den Verschlüsselungsprozess zu optimieren, wird einen Initialisierungsvektor (IV) zusammen mit dem symmetrischen Schlüssel verwendet. Ein IV ist ein Zufallswert, der zur Initialisierung des Verschlüsselungsalgorithmus verwendet wird, bevor die Daten verschlüsselt werden [158]. Der IV wird mit dem symmetrischen Schlüssel kombiniert, um einen eindeutigen Chiffrierschlüssel für jeden Datenblock zu erzeugen. Dadurch wird sichergestellt, dass selbst wenn dieselben Daten mehrmals mit demselben Schlüssel verschlüsselt werden, der resultierende Chiffriertext jedes Mal anders ist, was es für Angreifer schwieriger macht, die Ausgangsdaten zu dekodieren.

Um eine sichere und vertrauenswürdige Kommunikation zwischen den beteiligten Systemen sicherzustellen, werden moderne und standardisierte Sicherheitsprotokolle wie *Transport Layer Security* (TLS) und *Hypertext Transfer Protocol Secure* (HTTPS) verwendet, die eine verschlüsselte Datenübertragung ermöglichen [173].

4.5 Formale Darstellung

In den vorherigen Abschnitten wurden die Funktionen des entwickelten Systems im Rahmen einer Darstellung der drei Systemelemente – „Maschinenauswahl“, „Lizenzmodellauswahl“ und „Datensicherheit“ – beschrieben. In diesem Unterkapitel wird die Struktur und Funktionsweise des konzipierten sicheren Assistenzsystems anhand der vom Benutzer eingepflegten Informationen und der Interaktionen dieser Daten mit dem Assistenzsystem in einem Informationsmodell dargestellt. Hierfür werden ein Konzeptionsdiagramm und UML-Klassendiagramme samt dessen Notation verwendet [124].

Das konzeptionelle Diagramm dient der Darstellung des entwickelten Assistenzsystems und seiner Bestandteile, wodurch der Systemumfang veranschaulicht wird. Es fungiert als visuelles Hilfsmittel zur Erläuterung des Systemdesigns und trägt dazu bei, ein präzises Verständnis des Systems und seiner Komponenten zu vermitteln. Darüber hinaus unterstützt das konzeptionelle Diagramm bei der Identifizierung der wesentlichen Entitäten und deren Beziehungen. Das Konzeptionsdiagramm des entwickelten Systems ist in Abbildung 4-12 dargestellt. Diese Abbildung veranschaulicht das System zur Verwaltung von Laserbearbeitungsmaschinen sowie der damit verbundenen, vom Anwender erworbenen Technologien und Lizenzen. Es umfasst verschiedene Entitäten, einschließlich Laserbearbeitungsmaschinen, Technologien, Prozesstypen, Aufträge, Materialien, Organisationen, Nutzer, Technologiedaten, Lizenzen und Lizenzmodelle, die miteinander in Beziehung stehen.

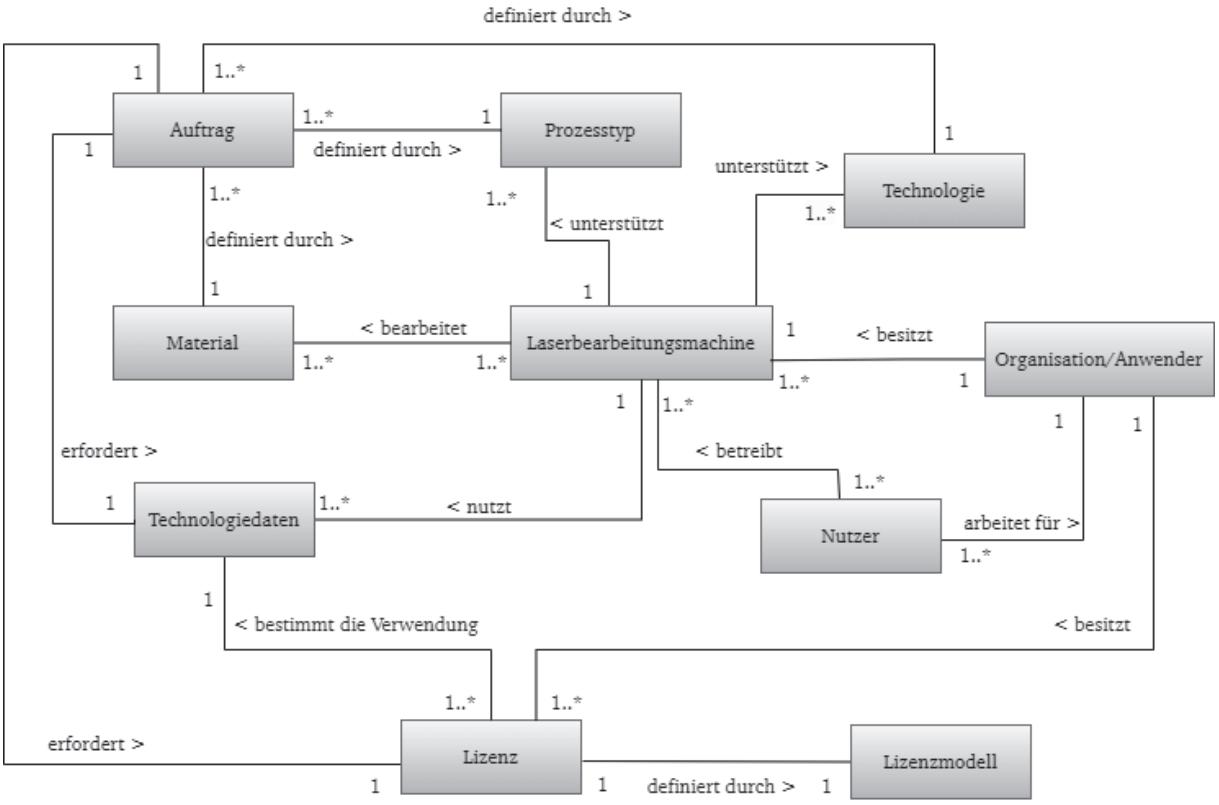

Abbildung 4-12: Konzeptionsdiagramm (eigene Darstellung)

Die Laserbearbeitungsmaschinen in diesem System unterstützen eine Vielzahl von Technologien, darunter 2-D, 3-D und Rohrverarbeitung, sowie verschiedene Prozesstypen wie Schneiden, Markieren und Schweißen. Diese Maschinen sind in der Lage, eine breite Palette von Materialien zu bearbeiten. Die Laserbearbeitungsmaschinen werden von den Nutzern oder Mitarbeitern einer Organisation betrieben, die mehrere dieser Maschinen besitzt. Die Organisation selbst stellt ein Anwender dar, der über mehrere Laserbearbeitungsmaschinen verfügt. Die Auftragsinformationen spielen eine entscheidende Rolle bei der Bestellung von Technologiedaten, die eine entsprechende Lizenz erfordern. Aufträge werden anhand verschiedener Kriterien definiert, darunter der Prozesstyp, die angewandte Technologie und die Art des zu bearbeitenden Materials. Wenn ein Auftrag bearbeitet werden soll, werden die benötigten Technologiedaten dem Anwender zur Verfügung gestellt und anschließend für die Durchführung des Laserbearbeitungsprozesses verwendet.

Lizenzen werden durch Lizenzmodelle definiert, die die Bedingungen für die Nutzung von Technologiedaten festlegen. Es existieren verschiedene Arten von Lizenzmodellen (siehe Kapitel 4.4.2), die dazu dienen, den Zugriff auf Technologiedaten zu regulieren und sicherzustellen, dass ausschließlich autorisierte Anwender diese Daten verwenden dürfen.

Das Informationsmodell fokussiert auf das zu untersuchende System. Es werden alle identifizierten und klassifizierten Informationen in einer formalen Struktur abgebildet. Die Klassen in einem Informationsmodell fassen ähnliche Objekte zusammen und bilden ihre Eigenschaften als Attribute ab. Die Zusammenhänge zwischen diesen Klassen werden zur Beschreibung der Systemprozesse als Beziehungen abstrahiert [159]. Aufgrund der sehr komplexen und umfangreichen Beschreibung der Gesamtthematik wird das Informationsmodell des entwickelten Assistenzsystems in mehrere Teilmodelle gegliedert, die im Folgenden vorgestellt werden.

Darstellung des formalen Modells vom Backend als UML-Klassendiagramm

Zunächst wird der Kern der Applikation, das Backend, modelliert. Das Backend ist die zentrale Komponente, die die verschiedenen Elemente des Systems miteinander verbindet und den Anwendern die benötigten Funktionen zur Verfügung stellt. Im Folgenden werden die Hauptaufgaben der Backend-Anwendung zunächst beschrieben. Das Backend ist zuständig für die Benutzerverwaltung, das umfasst unter anderem das Erstellen, Aktualisieren und Löschen von Nutzerkonten. Zusätzlich dient es der Authentifizierung und Autorisierung, womit das System vor Zugriff geschützt wird. Die Backend-Applikation setzt auf zwei unterschiedliche Authentifizierungsmethoden:

- Passwort und Autorisierungs-Token: Jedes Nutzerkonto ist mit einem Benutzernamen und einem Passwort ausgestattet. Bei der Anmeldung an der Benutzerschnittstelle muss der Nutzer diese Anmeldedaten eingeben. Bei erfolgreicher Anmeldung vergibt die Backend-Applikation ein Autorisierungs-Token an den Nutzer. Dieses Token wird vom Frontend bei jeder Anfrage an das Backend verwendet. Die Gültigkeit des Tokens bleibt bestehen, bis der Nutzer sich abmeldet oder nach einer vordefinierten Zeit der Inaktivität ausläuft.

- Verifizierungscode: Zum Schutz sensibler Systemfunktionen, wie zum Beispiel der Funktion zum Lizenzzugriff, fordert die Backend-Anwendung vom Nutzer die Eingabe seines Passworts sowie eines Verifizierungscodes, der an die E-Mail-Adresse des Nutzers gesendet wird. Dieses Verfahren, bekannt als Zwei-Faktor-Authentifizierung (2FA), bietet eine zusätzliche Sicherheitsebene zu dem herkömmlichen Benutzername-Passwort-Ansatz.

Das Backend bietet ebenfalls die Funktionen, Laserbearbeitungsmaschinen zu erstellen, zu aktualisieren und zu löschen. Zusätzlich erlaubt es das Einrichten neuer Laserfertigungsaufträge, für welche Technologiedaten und zugehörige Lizenzen benötigt werden.

Der Nutzer gibt die notwendigen Informationen ein, um einen neuen Auftrag zu starten, darunter Technologieart, Prozesstyp sowie Materialart und -dicke. Diese Informationen werden im ersten Systemelement, der „Maschinenauswahl“, verwendet, um die passenden Laserbearbeitungsmaschinen auszuwählen. Daraufhin sortiert das Backend die verfügbaren Maschinen aus und wählt daraus jene aus, die für die Bearbeitung des Auftrags geeignet sind.

Für die formale Darstellung des Backends werden zwei Pakete abgebildet, das Kern- (engl. Core) und das Authentifizierungspaket (engl. Authentication). Das UML-Klassendiagramm für das Backend wird in Abbildung 4-13 dargestellt. Die Abbildung zeigt die Module für die Verwaltung von Nutzern, Laserbearbeitungsmaschinen, Materialien und Laseraufträgen. Mit den definierten Klassen werden Objekte mit gemeinsamen Eigenschaften und derselben Semantik zusammengefasst. Diese Klassen weisen verschiedene Beziehungen auf – Assoziationsbeziehungen, Aggregationsbeziehungen sowie Kompositionsbeziehungen.

Nutzer werden durch grundlegende Informationen wie Name und E-Mail-Adresse sowie durch einzigartige Sicherheitsmerkmale wie Passwort und Token charakterisiert. Die Klasse *User* repräsentiert alle Nutzer des Systems und ihre Attribute wie Name, E-Mail-Adresse und Passwort. Diese Klasse ist mit der Klasse *Token* verbunden, die Attribute zur Autorisierung der Nutzer enthält. Die Multiplizität beträgt 1, sowohl auf der Seite des Tokens als auch auf der Seite des Nutzers. Während das Passwort zur Authentifizierung dient und dem Nutzer eine einmalige Anmeldung pro Sitzung ermöglicht, ist das Token ein Authentifizierungsschlüssel, der bei jeder Nutzeranfrage an das Backend mitgeschickt wird. Das Token unterstützt während einer Sitzung die Aufrechterhaltung der Verbindung zwischen dem Nutzer und dem System über mehrere Anfragen hinweg.

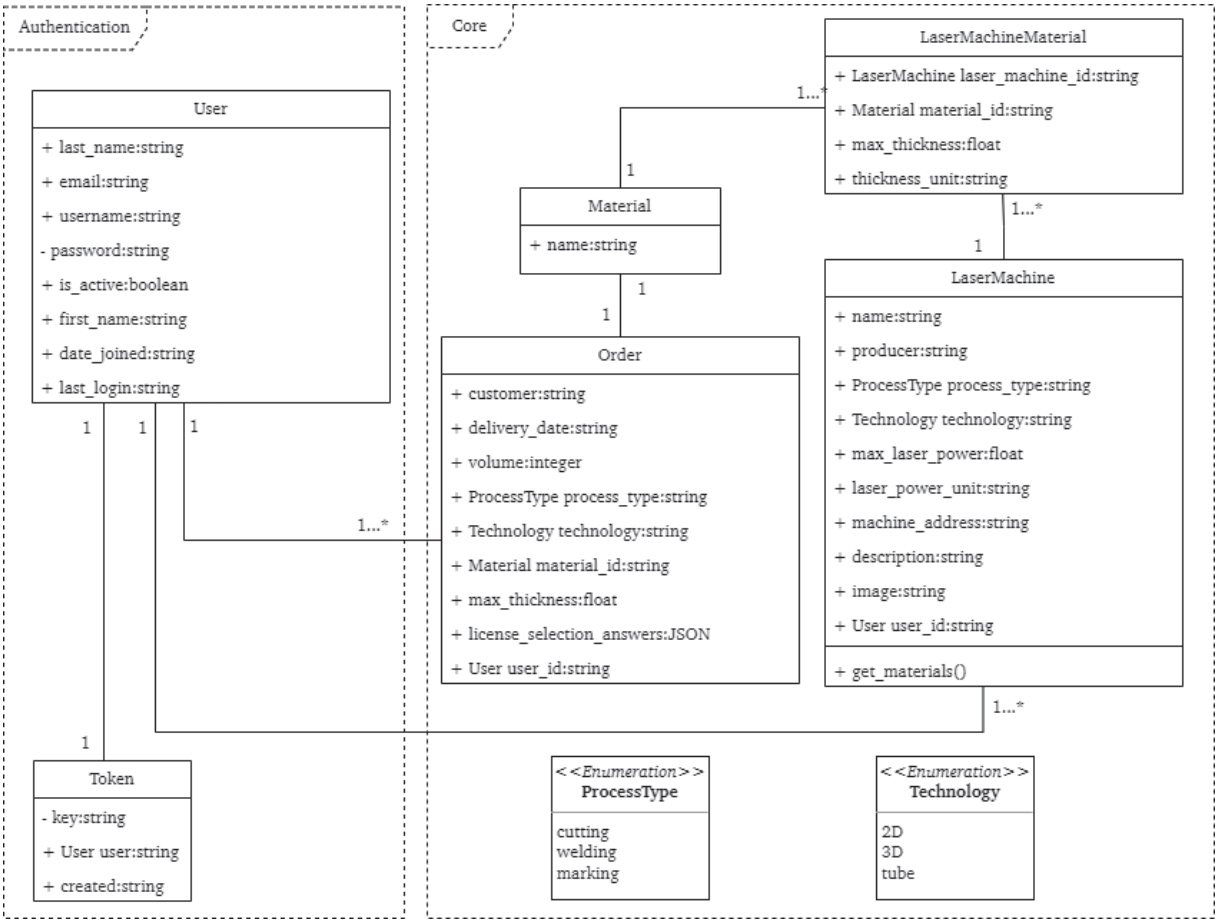

Abbildung 4-13: UML-Klassendiagramm des Backends (eigene Darstellung)

Im Core-Paketdiagramm sind vier Klassen definiert: *LaserMachine*, *LaserMachineMaterial, Material* und *Order*. Die Klasse *LaserMachine* speichert verschiedene Attribute von Laserbearbeitungsmaschinen, darunter die maximale Laserleistung, die Technologieart und den Prozesstyp. Ein besonders wichtiges Attribut in dieser Klasse ist *machine_address*. Dieses Attribut gibt die Adresse des Kryptosystems an, an die das Backend die Entschlüsselungsinformationen für die jeweilige Lizenz senden soll, wenn ein Nutzer die entsprechenden Technologiedaten nutzen möchte.

Da jede Lasermaschine mehrere Materialien bearbeiten kann und umgekehrt viele Materialien von verschiedenen Maschinen bearbeitet werden können, ist die Beziehung zwischen der Klasse *LaserMaschine* und der Klasse *Materialien* eine Many-to-Many-Beziehung. Um diese Beziehung deutlicher darstellen zu können, wird eine Klasse *LaserMachineMaterial* eingefügt. Diese Klasse verbindet jede Lasermaschine mit den Materialien, die sie bearbeiten kann. Außerdem speichert diese Klasse Informationen wie die maximale Dicke der jeweiligen Materialien, die eine Laserbearbeitungsmaschine bewältigen kann. Die Klasse *LaserMachineMaterial* fungiert somit

als Vermittler, indem sie die Beziehung zwischen der Klasse *LaserMachine* und der Klasse *Material* aufrechterhält und wichtige Informationen zur Kompatibilität speichert. Diese Eigenschaften können über die Operation *get_materials* abgerufen werden. Da jeder Nutzer *User* mehrere Aufträge *Orders* und Maschinen *LaserMachines* haben kann, wobei jeder Auftrag und jede Laserbearbeitungsmaschine genau einem Nutzer zugewiesen ist, steht die Klasse *User* in einer 1...*-Beziehung mit den Klassen *Order* und *LaserMaschine*.

Für die Auswahl der passenden Lizenzmodelle beantwortet der Nutzer zunächst eine Reihe von Fragen, die auf den spezifischen Auftragsinformationen und seiner bisherigen Erfahrung basieren. Diese Antworten werden an das Backend weitergeleitet. Dort transformiert die Backend-Anwendung die Nutzereingaben mithilfe einer vordefinierten Datenstruktur in Fakten, aus denen dann das Assistenzsystem des zweiten Elementes „Lizenzmodellauswahl“ das entsprechende Lizenzmodell ermittelt. Die Klasse *Order* ist für die Speicherung verschiedener Auftragsattribute verantwortlich, wie zum Beispiel den Kundennamen, das Lieferdatum und das Auftragsvolumen. Darüber hinaus speichert diese Klasse die Antworten des Benutzers auf Fragen, die zur Auswahl des geeigneten Lizenzmodells signifikant sind. Nachdem der Benutzer diese Fragen beantwortet hat, übermittelt das Backend diese an das Assistenzsystem, welches daraufhin das passende Lizenzmodell für den Auftrag ermittelt. Die Klasse *Order* spielt daher eine zentrale Rolle bei der Vereinfachung des Lizenzauswahlprozesses, da sie die zur Bestimmung des geeigneten Lizenzmodells erforderlichen Informationen speichert.

Das Backend bietet die Möglichkeit, alle vom Benutzer erworbenen Lizenzen und deren Zugriffshistorie zu erfassen und diese in einer Übersicht auf der Benutzerschnittstelle darzustellen. Darüber hinaus ermöglicht sie dem Benutzer, eine Anfrage zum Zugriff auf eine bestimmte Lizenz an den Lizenz-Vertrauensagenten zu senden. Der Benutzer muss dem Backend alle notwendigen Informationen für den Zugriff auf diese Lizenz zur Verfügung stellen. Sobald die Anfrage vom Lizenz-Vertrauensagenten genehmigt wurde, werden die Entschlüsselungsinformationen zusammen mit den Lizenzinformationen über den Lizenzmanager an das Backend gesendet. Das Backend leitet diese Informationen dann an das Kryptosystem der betreffenden Laserbearbeitungsmaschine weiter. Daher verfügt das Backend über Module, die die Integration weiterer Systemkomponenten wie das Assistenzsystem und den Lizenzmanager ermöglichen. Diese Integrationsmodule werden den nächsten Abschnitten vorgestellt.

Darstellung des formalen Modells vom Lizenz-Vertrauensagenten als UML-Klassendiagramm

Der Smart Contract ist ein Programm, das auf der Blockchain ausgeführt wird, um die Lizenzbedingungen umzusetzen. Es ist essentiell, das Datenmodell des intelligenten Vertrages zu definieren. Das Lizenzdatenmodell sollte alle notwendigen Informationen enthalten, die Technologiedatenmarktplätze benötigen, um die Nutzung der Technologiedaten zu regeln – die Lizenzmodelle unterscheiden sich hinsichtlich dieser Informationen. Zusätzlich sollte das Datenmodell die grundlegenden Metadaten enthalten, die von den verschiedenen Lizenzmodellen gemeinsam genutzt werden. Deshalb müssen die Arten von Lizenzmodellen, die erforderlichen Entschlüsselungsinformationen, die am Lizenzierungsprozess beteiligten Entitäten und die Regeln für die Erteilung und die Nutzung von Lizenzen analysiert werden. Die Entschlüsselungsinformationen enthalten jene Daten, die der Anwender benötigt, um die Technologiedaten decodieren zu können. Der Hersteller der Technologiedaten muss diese Informationen als verschlüsselte Zeichenfolge mit dem öffentlichen Schlüssel des Anwenders bereitstellen und sie mit seinem privaten Schlüssel signieren, um zu beweisen, dass diese Entschlüsselungsinformationen vom ihm erzeugt wurden. Auf diese Weise kann der Anwender die Authentizität der Entschlüsselungsinformationen überprüfen, indem er die digitale Signatur des Herstellers verifiziert. Außerdem kann nur der Anwender die Entschlüsselungsinformationen mit seinem privaten Schlüssel entschlüsseln, und somit wird die Vertraulichkeit der Daten sichergestellt. Diese Informationen werden im Lizenzdatenmodell in der Blockchain dargestellt. Der intelligente Vertrag hat das folgende Datenmodell, das in der Tabelle 4-1 dargestellt wird:

Tabelle 4-1: Lizenzdatenmodell

Daten	**Bezeichnung**	**Datentyp**
ID	Eindeutiger Identifikator für die Lizenz	String
Lizenzmodell	Die Bezeichnung des Lizenzmodells	String

Technologiedaten ID	Eindeutiger Identifikator für die Technologiedaten	String
Entschlüsselungsinformationen	Verschlüsselte Zeichenfolge des symmetrischen Schlüssels und des Algorithmus, der zur Ver-/Entschlüsselung der Technologiedaten verwendet wird	String
Hersteller	Name des Technologiedatenmarktplatzes, der die Lizenz ausgestellt hat	String
Anwender	Name des Anwenders, der die Lizenz erworben hat	String
Lizenzstatus	Der Status der Lizenz: aktiv oder inaktiv	Boolean
Aktivierungsdatum	Das Datum der ersten Aktivierung der Lizenz	String
Weitere Informationen	Weitere Informationen über das gewählte Lizenzmodell, z. B. die Ablaufzeit oder die Anzahl der erlaubten Nutzungen	String

Die in Abbildung 4-14 dargestellte Klassendarstellung veranschaulicht die Modellierung des Lizenz-Vertrauensagenten. Drei Klassen sind definiert: *LicenseModel, LicenseContract* und *OtherInformation*. Die Klasse *LicenseModel* fungiert als Repräsentation der Lizenzmodelle und dient zur Strukturierung der Lizenzdaten. Sie definiert die Repräsentation der Lizenzen auf der Blockchain. Diese Klasse strukturiert die Lizenzdaten, indem sie diese nach dem eindeutigen Identifikator *ID*, ihrem Namen *LicenseModel*, dem Lizenzstatus *Active* und dem Aktivierungsdatum *ActivationDate* unterscheidet. Außerdem enthält sie zusätzlich die Lizenznutzungsbedingungen wie das Ablaufdatum *Expiration* und die maximale Anzahl der Nutzungen *NumUsages*. Darüber hinaus enthält sie Informationen, die für die Nutzung der Technologiedaten benötigt werden, wie der eindeutige Identifikator der Technologiedaten

TechnologyDataID und die Entschlüsselungsinformationen *DecryptionInformation*. Weitere Eigenschaften bezüglich der Vertragspartei sind ebenfalls abgebildet wie die Eigenschaft *Manufacturer* für den Hersteller, der die Lizenz ausgestellt hat, und die Eigenschaft *User* für den Anwender, dem die Lizenz gehört.

Die erforderlichen Funktionalitäten zur Erstellung, Erteilung, Validierung und zum Widerruf von Lizenzen sowie die Regeln zur Aktualisierung der Lizenzangaben sollen durch den programmierten intelligenten Vertrag möglich sein.

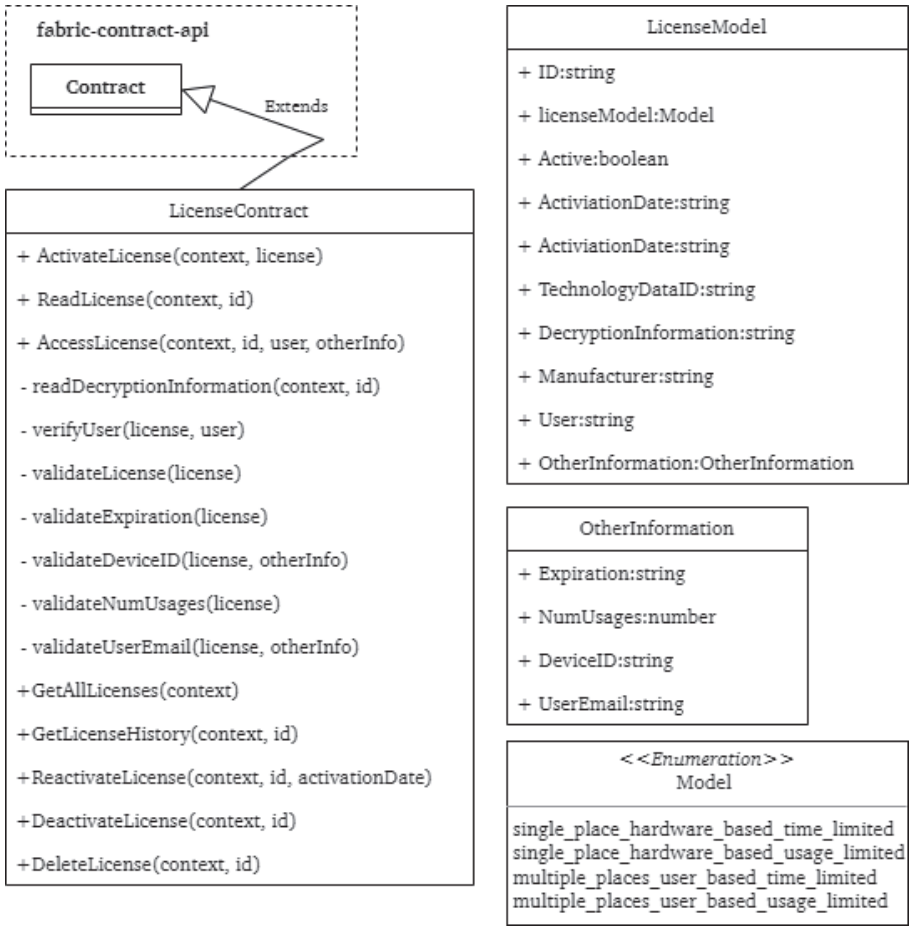

Abbildung 4-14: UML-Klassendiagramm des Lizenz-Vertrauensagenten (eigene Darstellung)

Die Klasse *LicenseContract* beinhaltet die für den Lizenzierungsprozess notwendigen Methoden. Sie bietet eine Reihe von Funktionen zur Verwaltung von Lizenzen. Außerdem enthält sie Funktionen für das Überprüfen von deren Gültigkeit und das Aufzeichnen der Zugriffshistorie. Mehrere Methoden sind universell und können sowohl vom Technologiedatenhersteller als auch vom Anwender angewendet werden, darunter beispielweise *GetAllLicenses*. Es gibt jedoch auch spezialisierte Methoden, die ausschließlich vom Anwender (wie z.B. *AccessLicense*) oder

vom Hersteller (wie z.B. *ActivateLicense*) ausgeführt werden können. Diese Funktionen können nur von autorisierten Anwendern aufgerufen werden. Die Funktion *ActivateLicense* ist für die Erstellung einer neuen Lizenz mit den angegebenen Nutzungsinformationen und die Bereitstellung dieser Lizenz in der Blockchain zuständig. Außerdem wird eine Aktivierungstransaktion mit den Entschlüsselungsinformationen erstellt, die für den Zugriff auf die Technologiedaten erforderlich sind. Ferner prüft diese Funktion vor der Aktivierung, ob die Lizenz bereits existiert, und validiert, ob die Nutzungsinformationen entsprechend des gewählten Lizenzmodells angegeben wurden, wie zum Beispiel das Ablaufdatum oder die maximale Anzahl der Nutzungen. Diese Funktion sollte vom Technologiedatenmarktplatz verwendet werden, um eine neue Lizenz zu erstellen, zu aktivieren und bereitzustellen.

Die Funktion *AccessLicense* prüft, ob ein Anwender berechtigt ist, auf eine Lizenz zuzugreifen, und ob die Lizenz noch gültig ist. Wenn eine nutzungsbasierte Lizenz noch gültig ist, wird die Anzahl der verbleibenden Nutzungen beispielweise um eine verringert und die Entschlüsselungsinformationen werden dem Anwender zugesandt. Diese Funktion muss vom Anwender jedes Mal aufgerufen werden, wenn er die Technologiedaten nutzen möchte. Dafür muss er sich zunächst authentifizieren. Die Funktion *GetAllLicenses* zeigt dem Anwender alle gespeicherten Lizenzen mit den dazugehörigen Informationen an. Die Funktion *GetLicenseHistory* zeigt dem Anwender die Zugriffshistorie einer Lizenz einschließlich aller früheren Versionen an. Weitere Funktionen zum Deaktivieren, Reaktivieren und Löschen von Lizenzen wurden ebenfalls implementiert.

Die Klasse *OtherInformation* stellt zusätzliche Lizenzinformationen für die Nutzungsbedingungen dar, wie die Gültigkeitsdauer der Lizenz (*Expiration*), die Anzahl der Nutzungen (*NumUsages*), die Maschinen-ID (*DeviceID*) und die E-Mail des Anwenders (*UserEmail*).

Die intelligenten Verträge unterstützen alle im Konzeptkapitel genannten Lizenzmodelle. In dieser Dissertation werden vier repräsentative Lizenzmodelle gewählt: *Einzelplatz, hardwarebasiert, zeitbegrenzt*; *Einzelplatz, hardwarebasiert, nutzungsbegrenzt*; *Mehrplatz, nutzerbasiert, zeitbegrenzt*; *Mehrplatz, nutzerbasiert, nutzungsbegrenzt*. Die Enumeration Model zeigt diese implementierten Lizenzmodelle an.

Die strukturierte Verwendung dieser Klassen und Funktionen gewährleistet eine granulare Kontrolle und Flexibilität innerhalb des Smart Contracts und ist essentiell für die Realisierung des Lizenzierungskonzeptes.

Darstellung des formalen Modells vom Lizenzmanager als UML-Klassendiagramm

Der Lizenzmanager übernimmt die Rolle eines Vermittlers zwischen dem Backend des Systems und dem Lizenz-Vertrauensagenten. Er muss in der Lage sein, bereitgestellte Lizenzen auf dem Lizenz-Vertrauensagenten aufzurufen und die benötigten Entschlüsselungsinformationen abzufragen. Danach leitet er die abgerufenen Lizenzen mittels einer definierten Schnittstelle an das Backend weiter. Das Klassendiagramm für den Lizenzmanager und das Integrationsmodul in das Backend ist in der Abbildung 4-15 dargestellt. Dabei werden zwei Pakete definiert *License Manager Proxy in The Backend Application* und *License Manager*.

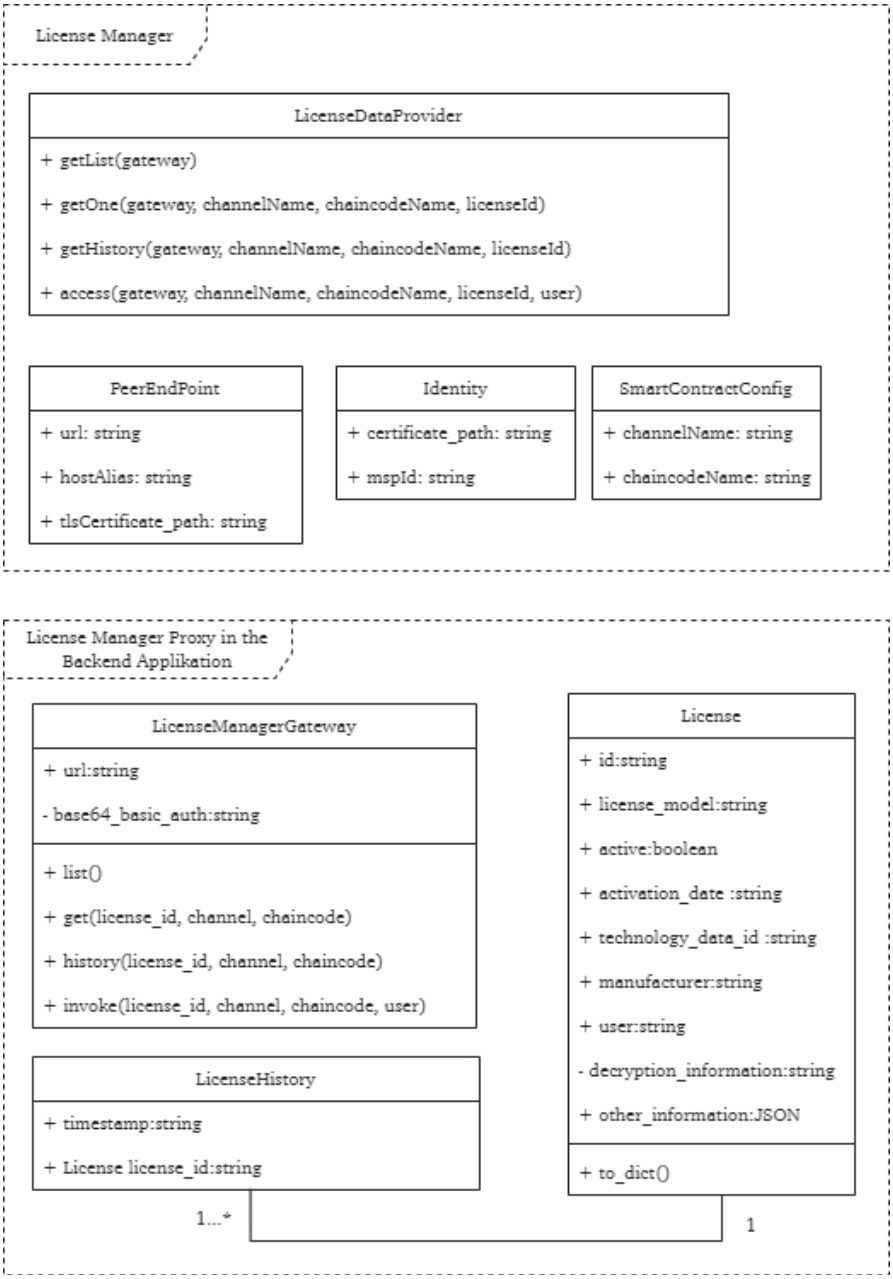

Abbildung 4-15: Klassendiagramm zum Lizenzmanager (eigene Darstellung)

Das Paket *License Manager* dient als Vermittler zwischen dem Lizenzmanager und dem Lizenz-Vertrauensagenten. Es umfasst Klassen, die für die Kommunikation verantwortlich sind, indem sie Anfragen an den Lizenz-Vertrauensagenten senden und Antworten von ihm empfangen. Hierfür werden essentielle Informationen benötigt, die in den Klassen *PeerEndPoint*, *Identity* und *SmartContractConfig* zusammengefasst werden. Die Klasse *LicenceDataProvider* beinhaltet die erforderlichen Methoden wie zum Beispiel die Operation *getList()* für das Abfragen von Informationen über alle existierenden Lizenzen auf dem Lizenz-Vertrauensagenten und die

Operation *getOne()* für den Bezug einer bestimmten Lizenz. Die übermittelten Informationen werden dann in ein für das Backend geeignetes Format umgewandelt.

Im Gegensatz dazu umfasst das Paket *License Manager Proxy in The Backend Application* die Kernfunktionen des Lizenzmanagers. Es beinhaltet Klassen, die eine Vielzahl von Aufgaben ausführen, einschließlich der Lizenzverwaltung, der Kommunikation mit der Backend-Applikation und der Bereitstellung von Schnittstellen für den Lizenzmanager-Proxy.

Im Paket *License Manager* sind die Klassen *Identity*, *PeerEndPoint, LicenseDataProvider* und *SmartContractConfig* definiert. Die Klasse *Identity* beinhaltet Attribute wie *certificate_path* und *mspId*, die für den Betrieb des Lizenz-Vertrauensagenten benötigt werden. Die Klasse *PeerEndPoint* speichert die erforderlichen Verbindungsinformationen, wie die Peer-Adresse (URL) und das TLS-Zertifikat. Die Klasse *LicenseDataProvider* im Lizenzmanager spielt eine entscheidende Rolle in der Interaktion zwischen dem Lizenzmanager und dem Lizenz-Vertrauensagenten. Sie stellt eine Reihe von Methoden zur Verfügung, die es ermöglichen, auf die verschiedenen Funktionen des Lizenzmanagers zuzugreifen und sie zu nutzen. Diese Methoden sind:

- *getList(gateway)*: Diese Methode wird angewandt, um eine Liste aller Lizenzen abzurufen, die der Benutzer erworben hat. Zudem werden sämtliche Informationen über diese Lizenzen ausgegeben.

- *getOne(gateway, channelName, chaincodeName, licenseId):* Diese Methode wird angewandt, um spezifische Informationen zu einer bestimmten Lizenz abzurufen. Die Eingabeparameter sind der Kanal- und Vertragsname sowie die ID der Lizenz.

- *getHistory(gateway, channelName, chaincodeName, licenseId):* Diese Methode wird angewandt, um die Zugriffshistorie einer bestimmten Lizenz zu ermitteln. Die Eingabeparameter sind der Kanal- und Vertragsname sowie die ID der Lizenz. Ausgegeben wird eine Liste der Zugriffsaktionen, die jeweils mit einem Zeitstempel versehen sind.

- *access(gateway, channelName, chaincodeName, licenseId, user)*: Diese Methode wird angewandt, um Zugriff zu erhalten zu einer bestimmten Lizenz oder Entschlüsselungsinformationen für Technologiedaten anzufordern. Die

Eingabeparameter sind der Kanal- und der Vertragsname sowie die ID der entsprechenden Lizenz und des Nutzers.

Die Klasse *SmartContractConfig* fasst Eigenschaften zum intelligenten Vertrag zusammen, einschließlich der Attribute *channelName* und *chaincodeName*, welche den Namen des Kanals und den Namen des Vertrags repräsentieren, auf dem bzw. unter dem der intelligente Vertrag gespeichert wird.

Das Paket *License Manager Proxy in the Backend Application* umfasst mehrere Klassen, um Informationen über Lizenzen und Zugriffshistorien zu verarbeiten. Die Klasse *LicenseManagerGateway* ist dabei zentral und stellt Methoden zur Verfügung, um mit dem Lizenzmanager zu interagieren und dessen API zu nutzen. Diese Methoden umfassen:

- *list():* Eine Methode zum Auflisten aller erworbenen Lizenzen.
- *get():* Eine Methode zum Abrufen spezifischer Lizenzinformationen.
- *history():* Eine Methode zum Abrufen der Zugriffshistorie einer bestimmten Lizenz.
- *invoke():* Eine Methode zum Anfragen des Zugriffs auf eine bestimmte Lizenz bzw. zum Anfragen der Entschlüsselungsinformationen für die verschlüsselten Technologiedaten.

Die Klasse *License* enthält eine Vielzahl von Informationen zu den erworbenen Lizenzen. Dies beinhaltet die eindeutige Identifikationsnummer der Lizenz *Id*, die Identifikationsnummer der Technologiedaten *technology_data_Id*, auf die die Lizenz Zugriff gewährt. Des Weiteren ist der Name des Herstellers *manufacturer*, der die Lizenz bereitgestellt hat, und des Benutzers *user*, der die Lizenz erworben hat, enthalten. Zudem gibt das Lizenzmodell *license_model* die Nutzungsbedingungen der Lizenz vor und die Entschlüsselungsdaten *decryption_information* sind die Informationen, die zur Entschlüsselung der Technologiedaten benötigt werden. Mit all diesen Informationen kann das Backend die erworbenen Lizenzen der Benutzer verwalten, den Zugriff auf die durch die Lizenzen zugänglichen Technologiedaten regeln und die Einhaltung der Nutzungsbedingungen des Lizenzmodells überprüfen. In Bezug auf die Beziehungen zwischen den Klassen zeigt die Multiplizität auf der Seite der Klasse *License* eine 1…*-Beziehung zur Klasse *LicenseManagerGateway* an. Dies bedeutet, dass einem einzelnen Lizenzobjekt mehrere Zugriffsaktionen zugeordnet werden können, die jeweils durch unterschiedliche Zeitstempel gekennzeichnet sind. Das ermöglicht eine genaue Nachverfolgung und

Dokumentation aller Zugriffe auf eine bestimmte Lizenz. Die Backend-Applikation stellt eine Verbindung mit dem Lizenzmanager her, um die oben genannten Funktionen zu nutzen und sie dem Nutzer über das Frontend zur Verfügung zu stellen.

Zusammen ermöglichen diese beiden Paketdiagramme eine effektive Kommunikation zwischen dem Backend und dem Lizenzmanager und gewährleisten, dass das Backend die erforderlichen Lizenzen vom Lizenzmanager abrufen kann. Darüber hinaus erlaben sie das Abrufen von Entschlüsselungsdaten sowie von lizenzbezogenen Informationen vom Lizenz-Vertrauensagenten.

Darstellung des formalen Modells vom Kryptosystem

Das Kryptosystem ist eine Software, das auf dem Server, auf spezieller Hardware oder direkt auf der Laserbearbeitungsmaschine installiert werden kann. Diese kann alleinstehend laufen oder in die Laserbearbeitungsmaschine integriert werden. Das Kryptosystem ist für die Entschlüsselung der verschlüsselten Technologiedaten zuständig. Auf Basis der Technologiedaten-ID in den Lizenzinformationen ruft es die entsprechenden Technologiedaten ab, die bereits auf einem gemeinsamen Ordner innerhalb des Netzwerks abgelegt sind. Hierfür als Entschlüsselungsinformationen der verschlüsselte symmetrische Schlüssel und der Verschlüsselungsalgorithmus benötigt. Diese werden mittels eines TPM (*Trusted Platform Module*) entschlüsselt. Das TPM ist eine spezialisierte Hardwarekomponente, die eine sichere Umgebung für die Speicherung und Verarbeitung sensibler Daten wie kryptografische Schlüssel, digitale Zertifikate und Passwörter zur Verfügung stellt. TPM-Komponente wird in der Regel in Personalrechner, Server und andere Computersysteme integriert, um hardwarebasierte Sicherheitsfunktionen anzubieten. Das TPM arbeitet unabhängig vom Betriebssystem und verfügt über eine eigene Firmware und einen eigenen Software-Stack. Es bietet eine Reihe von Sicherheitsmechanismen wie Verschlüsselung, digitale Signaturen und sicheres Hochfahren, um die Integrität des Systems und die Sicherheit der Daten zu gewährleisten [160]. Darunter fallen auch die Generierung von Zufallszahlen und Schlüsseln sowie die Speicherung von Schlüsseln und die Attestierung, die es Anwendern ermöglicht, die Authentizität eines Systems oder einer Komponente zu überprüfen. Die Abbildung 4-16 zeigt das Klassendiagramm des Kryptosystems, das aus vier Klassen besteht: *Server, TechnologyDataVendor, TPM_Gateway* und *Settings*. Die Klasse *Server* ist mit der Operation *do_POST* ausgestattet, welche zur Bearbeitung von HTTP-POST-Anfragen dient. In diesem Kontext liest und verarbeitet diese Methode Anfragedaten, die in der Form eines JSON-Objekts vorliegen, und enthält eine Lizenz, die eine kryptographisch verschlüsselte Technologiedaten-ID und die dazugehörigen Entschlüsselungsinformationen

umfasst. Parallel dazu bietet die Klasse *TechnologyDataVendor* spezialisierte Methoden zur Entschlüsselung und Überprüfung der digitalen Signatur der kryptographisch verschlüsselten Technologiedaten. Als Eingabe verwendet diese Klasse eine *ID*, die den Technologiedaten zugeordnet ist, und verschlüsselte Entschlüsselungsinformationen mit einer zugehörigen digitalen Signatur.

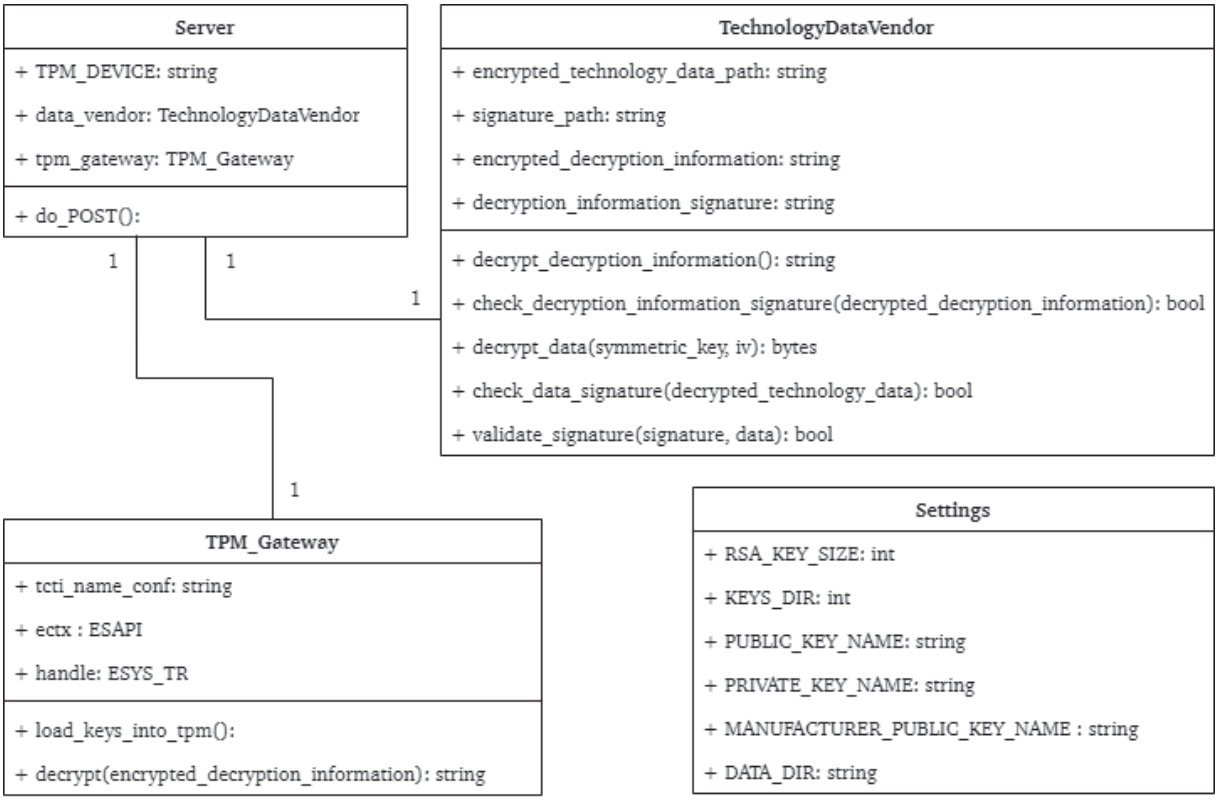

Abbildung 4-16: Klassendiagramm des Kryptosystems (eigene Darstellung)

Im Kontext der Kryptosystem-Funktionalität spielt die *Settings*-Klasse eine zentrale Rolle, da sie die Konfigurationseinstellungen repräsentiert und unter anderem Details zu den öffentlichen und privaten Schlüsseln des Benutzers speichert. Ein besonderes Attribut, das *DATA_DIR* genannt wird und einen Zeichenkettenwert speichert, verweist auf den Speicherpfad der verschlüsselten Technologiedaten. Abschließend dient die *TPM_Gateway*-Klasse als Schnittstelle zum TPM und ermöglicht das Laden des privaten Schlüssels in das TPM und die Entschlüsselung der kryptographisch verschlüsselten Schlüsseldaten. Die Klasse *TPM_Gateway* bildet die Funktionalität des TPMs ab. Der Klassenkonstruktor initialisiert den TPM-Kontext. Die *decrypt*-Methode verwendet eine base64- kodierte Zeichenkette mit den verschlüsselten Entschlüsselungsinformationen, die dann mit dem in das TPM eingelegten privaten Schlüssel entschlüsselt werden.

Darstellung des formalen Modells vom wissensbasierten Assistenzsystem

Um das wissensbasierte Assistenzsystem mit dem Backend zu integrieren, müssen bestimmte Entitäten definiert werden. Diese Entitäten ermöglichen es, die Antworten des Nutzers auf die vordefinierten und gestellten Fragen über die Benutzerschnittstelle zu sammeln und sie der Inferenzmaschine des Assistenzsystems in einer interpretierten Weise zur Verfügung zu stellen. Darüber hinaus sollten Entitäten definiert werden, die das Ergebnis des Inferenzprozesses, d. h. die ausgewählten Lizenzmodelle, dem Nutzer anzeigen können. Diese Integration ist entscheidend, um das Assistenzsystem effektiv in der Backend-Umgebung der Anwendung zu nutzen. Das Integrationsmodell wird in der Abbildung 4-17 dargestellt. Im Modell werden Klassen und Enumerationen definiert.

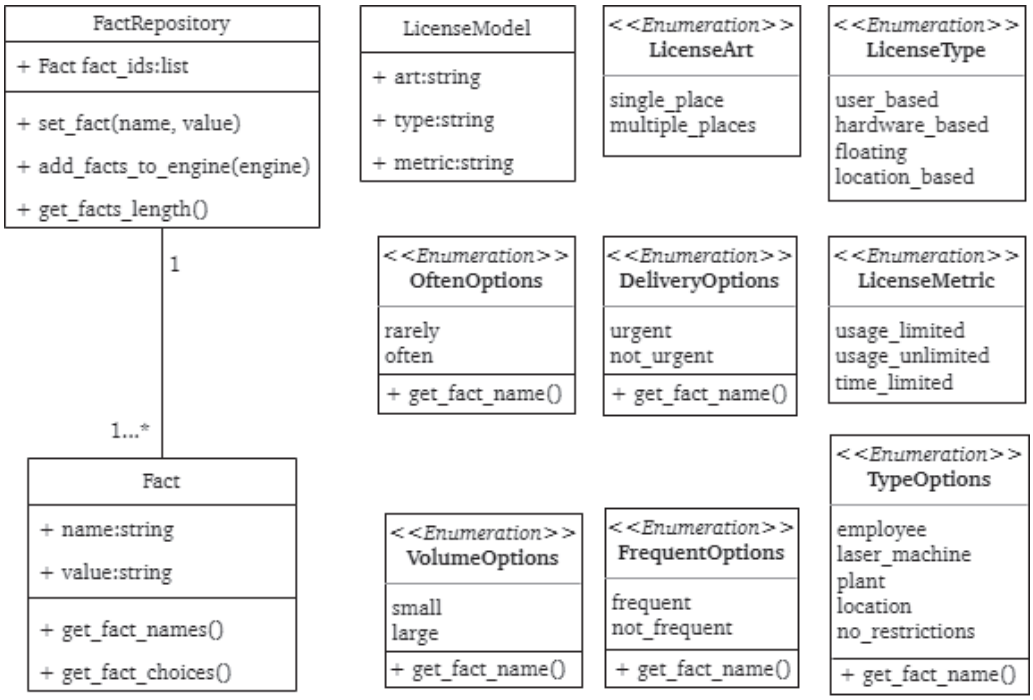

Abbildung 4-17: Integrationsmodell des Assistenzsystems ins Backend (eigene Darstellung)

Die Klasse *LicenseModel* repräsentiert die verschiedenen Lizenzmodelle, unter denen das Assistenzsystem auswählen kann. Sie enthält Attribute wie *art, type* und *metric*, die jeweils die Art, den Typ und die Metrik eines Lizenzmodells repräsentieren. Die Klasse *FactRepository* dient als Sammlung von Fact-Objekten. Mithilfe der Methode *set_fact* kann ein neues Fact-Objekt zur Sammlung hinzugefügt werden. Die Methode *add_facts_to_engine* ermöglicht die Übermittlung aller gesammelten Fakten an eine spezifische Inferenzmaschine. Darüber hinaus stellt die Methode *get_facts_length* die Gesamtzahl der gespeicherten Fakten dar.

Die Klasse *Fact* stellt alle definierten Fakten dar, die mit Auswahlkriterien zusammenhängen. Sie hat zwei Attribute, *name* und *value*, die den Namen und den Wert des Fakts repräsentieren. Die Klasse bietet zudem zwei Methoden, *get_fact_names* und *get_fact_choices*, die eine Liste aller Faktnamen bzw. die Auswahlmöglichkeiten für jeden Fact liefern. In Bezug auf die Beziehung zwischen diesen Klassen ist die Klasse *Fact* in einer 1...*-Beziehung mit *FactRepository*, was bedeutet, dass ein *FactRepository*-Objekt eine oder mehrere Fact-Objekte enthalten kann, während ein *Fact*-Objekt immer genau zu einem *FactRepository*-Objekt gehört.

Die Enumeration *LizenzArt* beinhaltet zwei Optionen: *single_place* und *multiple_place,* die festlegen, ob die Lizenz an einer einzigen Maschine oder an mehreren Maschinen verwendet werden kann. Die Enumeration *LicenseType* umfasst vier Optionen: *user_based, hardeware_based, floating und location_based*. Diese Optionen definieren, auf welcher Basis und mit welchen Einschränkungen die Lizenz bereitgestellt wird – beispielsweise basierend auf der Anzahl der Benutzer, der verwendeten Hardware, der Möglichkeit der gemeinsamen Nutzung der Lizenz oder des Standorts der Lizenznutzung. Die Enumeration *LicenseMetric* definiert drei Möglichkeiten für die Art der Lizenzmetrik: *usage_limited, usage_unlimited* und *time_limited*. Diese Optionen bestimmen, wie die Nutzung der Lizenz gemessen und eingeschränkt wird. Die Enumerationen *DeliveryOptions*, *TypeOptions*, *VolumeOptions*, *OftenOptions* und *FrequentOptions* repräsentieren die verschiedenen Auswahlmöglichkeiten für festgelegte Auswahlkriterien wie Lieferzeit, Auftragsvolumen und -häufigkeit. Jede dieser Enumerationen enthält eine Methode *get_fact_name*, die den Namen des Fakts bereitstellt, der mit dieser bestimmten Enumeration verbunden ist. Die Verwendung dieser Enumerationen erleichtert die Arbeit mit dem Assistenzsystem, indem sie eine klar definierte Menge an Optionen bereitstellen, die dann vom Assistenzsystem zur Auswahl der geeigneten Lizenzmodelle verwendet werden können.

4.6 Fazit zum Konzept

In dieser Dissertation wird ein effektives und sicheres Assistenzsystem entwickelt, das den Anwender bei der Nutzung von Technologiedatenmarktplätzen am Beispiel der Lasermarkierung unterstützt. Das Assistenzsystem soll Handlungsempfehlungen für den Bezug von Technologiedaten von unterschiedlichen Marktplätzen liefern. Im Rahmen des vorliegenden Konzepts wird ein systematisches und auftragsspezifisches Verfahren für die

Maschinen- und Lizenzmodellauswahl und zur sicheren Übermittlung der Technologiedaten und der zugehörigen Lizenzen vorgestellt.

Ausgehend von dem in Kapitel 3 definierten Anforderungsprofil wurde in diesem Abschnitt ein Konzept zur Unterstützung des Anwenders bei der Nutzung von Technologiedatenmarktplätzen entwickelt und vorgestellt. Die Hauptbausteine dieses Konzeptes bilden die Systemelemente „Maschinenauswahl“, „Lizenzmodellauswahl“ und „Datensicherheit“. Dieses Konzept kommt im Rahmen der Planungs- und Produktionsphase des Laserbearbeitungsprozesses zum Tragen. Eine Gesamtübersicht über das Konzept, bestehend aus den entwickelten Systemelementen, wurde in Unterkapitel 4.3 gegeben. Diese Übersicht wurde durch die Vorstellung der Systemelemente ergänzt.

In Kapitel 4.4.1 wurde die systematische Auswahl einer geeigneten Laserbearbeitungsmaschine unter Berücksichtigung der Prozess-, Technologie- und Materialeigenschaften konzipiert. Die Auswahl erfolgt schnell, sie ist rechnerbasiert und ermöglicht dem Anwender eine effiziente Bearbeitung von Kundenaufträgen. Hierfür wurden die Auswahlattribute definiert und die Schritte des Auswahlprozesses wurden vorgestellt.

Im Unterkapitel 4.4.2 wurde das Systemelement „Lizenzmodellauswahl“ konzipiert, das dem effektiven und flexiblen Erwerb von Technologiedaten für die gewählten Laserbearbeitungsmaschinen dient. Dafür wurden zunächst passende Lizenzmodelle für die Technologiedatenvermarktung vorgestellt. Danach wurden die Attribute der Lizenzmodelle sowie deren Merkmalwerte, die den Bedürfnissen des Anwenders entsprechend flexibel bestimmt werden können, definiert. Das ermöglicht einen einfachen, effizienten und systematischen Auswahlprozess bezüglich der Lizenzmodelle. Zu diesem Zweck wurde ein wissensbasiertes Assistenzsystem basierend auf Methoden der künstlichen Intelligenz und der Entscheidungsbäumen entwickelt. Jeder Zweig dieses Entscheidungsbaums entspricht einer klaren und präzisen Frage, die dem Benutzer gestellt wird, um systematisch das passende Lizenzmodell zu empfehlen.

Um den Anforderungen hinsichtlich der Datensicherheit gerecht zu werden, wurde das Systemelement „Datensicherheit“ in das Konzept integriert. In diesem Zusammenhang wurde das Lizenzierungskonzept im Hinblick auf den Erwerb und die Nutzung von Technologiedaten vorgestellt. Beginnend mit der Generierung einer Lizenz auf dem Technologiedatenmarktplatz bis zur Nutzung dieser Lizenz durch den Anwender wurden die Prozessschritte sowie die benötigten Systeme vorgestellt. Es wurden drei Subsysteme eingeführt: der Lizenz-

Vertrauensagent, der Lizenzmanager und das Kryptosystem. Schließlich wurde die technische Seite des Sicherheitskonzepts beschrieben.

In Unterkapitel 4.5 wurde dargelegt, welche Informationen der Nutzer in das entwickelten System einspeisen muss, welche Schritte der Prozess der sicheren Beschaffung von Technologiedaten umfasst und wie der Datenaustausch zwischen den Systemelementen vor sich geht. Hierfür wurden entsprechende konzeptionelle Informationsmodelle als UML- Klassendiagramme entwickelt. Ferner wurden die vorgestellten Subsysteme anhand der UML- Klassendiagrammen modelliert.

Die Wahl eines systematischen, flexiblen und auswahlorientierten Ansatzes und dessen Darstellung im Rahmen einer Informationsmodellierung hat sich in diesem Kapitel als vorteilhaft erwiesen. Ausgehend von der rechnerbasierten Auswahl der passenden Laserbearbeitungsmaschinen und Lizenzmodelle konnte eruiert werden, welche Informationen für den effektiven Bezug von Technologiedaten erforderlich sind. Die hier erfolgte Inblicknahme technischer Besonderheiten der Nutzung von Technologiedaten begünstigt zudem die Erfüllung gängiger Sicherheitsanforderungen. Im Rahmen der formalen Abbildung der eben erwähnten Informationen wurden letztlich Informationsmodelle für die prototypische Implementierung eines effektiven und sicheren Assistenzsystems für die Beschaffung von Technologiedaten auf entsprechenden Marktplätzen am Beispiel der Lasermarkierung erstellt. Im nachfolgenden Kapitel erfolgt die prototypische Implementierung dieses Assistenzsystems.

5 PROTOTYPISCHE IMPLEMENTIERUNG

Das in dieser Dissertation betrachtete System für die Unterstützung des Anwenders bei der effektiven und sicheren Nutzung von Technologiedatenmarktplätzen in der Lasermarkierung wurde bereits in den vorherigen Kapiteln analysiert, spezifiziert und entwickelt. Nun wird es prototypisch implementiert. Die Implementierung umfasst daher die Entwicklung einer Webapplikation sowie eines privaten Blockchain-Netzwerks für die Bereitstellung und sichere Nutzung von Technologiedaten in der Lasermarkierung. Hierbei werden Funktionselemente zur Unterstützung des Anwenders bei der Maschinen- und Lizenzmodellauswahl in der Planungsphase implementiert. Außerdem wird das Konzept zur Datensicherheit und Nutzungsverfolgung von erworbenen Technologiedaten und der zugehörigen Lizenzen in der Produktionsphase umgesetzt.

Basierend auf den in Kapitel 3 festgelegten Anforderungen an die Implementierung und den Erläuterungen in Kapitel 4 folgt nun die Beschreibung der Implementierung eines sicheren Assistenzsystems zur effektiven Nutzung von Technologiedatenmarktplätzen in der Lasermarkierung. Vor der Vertiefung in die Einzelheiten der Implementierung werden folgende Ausgangspunkte erläutert:

- Das Erwerben von Technologiedaten und Lizenzen wird beim Anwender durch die Enterprise-Resource-Planning-Abteilung (ERP) organisiert.

- Der Prozess des Erwerbens und der Generierung von Lizenzen wird vonseiten des Technologiedatenmarktplatzes ausgeführt.

- Im Falle eines erfolgreichen Erwerbs von lizenzierten Technologiedaten wird ein Smart Contract erstellt und auf dem Lizenz-Vertrauensagenten innerhalb der Blockchain hinterlegt. Dieser Smart Contract repräsentiert die vereinbarten Nutzungsbedingungen des ausgewählten Lizenzmodells und beinhaltet die notwendigen Informationen für die Zuordnung und Verfolgung von deren Nutzung.

- Dem Käufer bzw. Anwender wird ein Lizenz-Token (eine eindeutige Identifikationsnummer) zugeordnet. Diese Identifikationsnummer ist essentiell für die

korrekte Zuweisung der erworbenen Lizenzen zu den jeweiligen Nutzern und Technologiedaten.

- Ab diesem Punkt kann der Anwender jederzeit eine Anfrage an den Lizenz-Vertrauensagenten stellen, wenn er die Technologiedaten auf den Laserbearbeitungsmaschinen verwenden möchte. Nach der Nutzung werden die Informationen bezüglich des Lizenznutzungsstatus aktualisiert und in der Blockchain auf dem Lizenz-Vertrauensagenten verzeichnet. Diese Informationen sind sowohl für den Anwender als auch für den Technologiedatenmarktplatz zugänglich.

Ob das System sinnvollerweise als Web- oder als Desktopanwendung entwickelt wird, ist abhängig von verschiedenen Faktoren, darunter spezifische Projekterfordernisse, die Anforderungen des Systems als auch die Präferenzen der Benutzer. Eine Webapplikation ist eine Softwareapplikation, die auf einem Webserver läuft und über einen Webbrowser wie *Google Chrome*, *Firefox* oder *Safari* aufgerufen wird. Eine Webapplikation wird nicht wie eine Desktopanwendung auf dem Rechner des Anwenders installiert [161]. Ein wesentlicher Vorteil von webbasierten Anwendungen ist die Möglichkeit eines Netzwerkzugriffs von jedem beliebigen Rechner, was insbesondere bei Benutzerinteraktionen aus der Ferne nützlich ist. Webapplikationen sind also von verschiedenen Orten zugänglich, sie können aber auch einfacher aktualisiert werden und sind wartungsfreundlich. Sie bieten zudem eine hohe Flexibilität bei der Verwendung durch diverse Nutzer auf unterschiedlichen Rechnern. [161]. Im Gegensatz dazu können desktopbasierte Anwendungen eine erhöhte Leistungsfähigkeit bieten, indem sie die volle Rechenkraft des Nutzerrechners nutzen. Darüber hinaus verfügen Desktop-Anwendungen häufig über erweiterte Möglichkeiten für benutzerdefinierte Anpassungen und bieten in der Regel eine umfangreichere und intuitivere Benutzeroberfläche im Vergleich zu Web-Applikationen [162]. Das vorgeschlagene System erfordert eine Interaktion zwischen unterschiedlichen Komponenten und sollte für Nutzer an verschiedenen Arbeitsplätzen über ein Netzwerk zugänglich sein. Deshalb erfolgt die Implementierung des sicheren Assistenzsystems in dieser Dissertation über die Entwicklung einer Webapplikation, um von den bereits genannten Vorteilen dieser Variante zu profitieren.

Zu Beginn dieses Kapitels werden die Systemarchitektur der technischen Umsetzung und ihre einzelnen Komponenten dargelegt. Auf dieser Grundlage erfolgt die Darstellung der Implementierung und Integration der Systemkomponenten in das Gesamtsystem. Diese systematische Herangehensweise ermöglicht ein tiefgehendes Verständnis der funktionalen und

strukturellen Perspektiven des Assistenzsystems und stellt sicher, dass die entwickelte Webapplikation den Anforderungen optimal entspricht.

5.1 Systemarchitektur

Das entwickelte Assistenzsystem dient der Unterstützung des Anwenders bei der effektiven Bereitstellung und der sicheren Nutzung von erworbenen Technologiedaten in der Lasermarkierung. In Kapitel 4.4 wurden die Systemelemente zur Unterstützung des Anwenders bei der Nutzung von Technologiedatenmarktplätzen in der Lasermarkierung entwickelt und beschrieben. Die Wissensgrundlage für die Implementierung der Webapplikation stellen die zuvor in Kapitel 4.5 entwickelten Informationsmodelle dar.

Eine Systemarchitektur ist ein Entwurf einer Softwareanwendung auf einem hohen Abstraktionsniveau. Sie präsentiert die Gesamtstruktur des entwickelten Systems, umfasst seine Subsysteme und bildet deren Interaktionen ab [163]. Ein System wird in der ISO/IEC 15288, (Systems and software engineering- System Life Cycle Processes) folgendermaßen definiert:

> ***System is a combination of interacting elements organized to achieve one or more stated purposes [185].***

Eine Systemarchitektur wird in der ISO/IEC/IEEE 42010 (Systems and software engineering - Architecture description) wie folgt definiert:

> ***Architecture of a system is defined as fundamental concepts or properties of a system in its environment embodied in its elements, relationships, and in the principles of its design and evolution [165].***

Nun wird die passende Systemarchitektur für die technische Umsetzung der Webapplikation gewählt. Es existieren diverse Systemarchitekturen, die für die Entwicklung einer Applikation in Betracht gezogen werden können, darunter monolithische, mikroservice- und serviceorientierte sowie ereignisgesteuerte Architekturen. Die Auswahl hängt von den spezifischen Bedürfnissen und Anforderungen der jeweiligen Applikation ab. Die monolithische Architektur (engl. Monolithic Architecture) charakterisiert sich dadurch, dass die gesamte Applikation als eine autonome, in sich geschlossene Einheit entwickelt wird [166]. Dies vereinfacht den Entwicklungsprozess, kann jedoch zu Herausforderungen führen, wenn die

Applikation gewartet oder skaliert werden muss. Im Gegensatz dazu wird bei der Mikroservice-Architektur (engl. Microservices Architecture) die Applikation in kleine, unabhängige Dienste zerlegt. Dies erlaubt eine größere Flexibilität und einfache Skalierung, erfordert jedoch eine sorgfältige Planung und Koordination der Dienste. Die serviceorientierte Architektur (SOA) (engl. Service-oriented Architecture) konzipiert die Applikation als eine Sammlung von Diensten, die über eine gemeinsame Schnittstelle miteinander kommunizieren. Die SOA fördert die Wiederverwendung und Modularität, kann jedoch komplex sein in der Umsetzung. Die ereignisgesteuerte Architektur (engl. Event-driven Architecture (EDA)) strukturiert die Applikation basierend auf einer Reihe von Ereignissen oder Nachrichten, die Aktionen in verschiedenen Komponenten des Systems auslösen. EDA bietet hohe Reaktionsfähigkeit und Flexibilität, kann aber die Debugging- und Fehlerbehebungsprozesse erschweren. Monolithische Architekturen haben den Vorteil, dass sie vergleichsweise einfach zu entwickeln und bereitzustellen sind. Ihre Skalierbarkeit und Modifizierbarkeit stellen jedoch große Herausforderungen dar [166]. Sowohl die serviceorientierte Architektur (SOA) als auch die ereignisgesteuerte Architektur (EDA) bieten eine größere Flexibilität als monolithische Architekturen, doch deren Entwicklung und die Wartung sind komplexer. Die Microservices-Architektur wurde als Lösung für die Beschränkungen traditioneller monolithischer Architekturen entwickelt, die mitunter mit dem Wachstum der Anwendungsentwicklung zunehmend komplex und schwer zu warten sind. Sie teilt die Anwendung in kleinere, leichter handhabbare Komponenten auf, was die Skalierbarkeit und Modifizierung einzelner Teile der Applikation erleichtert, ohne den gesamten Systemaufbau zu beeinflussen. Die Microservices-Architektur bietet eine Reihe von Vorteilen, darunter verbesserte Skalierbarkeit, erhöhte Fehlertoleranz, verkürzte Markteinführungszeit sowie vereinfachte Wartung und Aktualisierung [166]. Dieser Ansatz ermöglicht es den Entwicklern, jeden Applikationsteil als Microservice unabhängig von anderen Teilen zu entwickeln, zu testen und bereitzustellen, ohne den Aufbau des gesamten Systems zu beeinträchtigen. Vor diesem Hintergrund wird die Webapplikation in dieser Dissertation unter Verwendung der Microservices-Architektur entwickelt.

Das Assistenzsystem wird hierbei als eine Zusammenstellung mehrerer Mikroservices implementiert – Abbildung 5-1 illustriert die Architektur des gesamten Systems. Jeder dieser Mikroservices wurde zur Erfüllung eines spezifischen Dienstes entwickelt und ist für seine eigene Datenspeicherung, -verarbeitung und -präsentation zuständig. Die Kommunikation zwischen den Mikroservices erfolgt über Anwendungsprogrammierschnittstellen (engl. Application Programming Interfaces (APIs)). Diese Architektur fördert die Dezentralisierung

und die Unabhängigkeit der einzelnen Dienste, erhöht die Skalierbarkeit und verbessert die Fehlertoleranz des gesamten Systems.

Die Architektur des Gesamtsystems umfasst verschiedene Komponenten: den Lizenz-Vertrauensagenten, den Lizenzmanager, das Backend, das Frontend, das Kryptosystem sowie das Assistenzsystem. Diese Komponenten können entweder auf einem einzelnen Rechner oder auf mehreren Maschinen innerhalb des Netzwerks des Unternehmens bereitgestellt werden.

Der Lizenz-Vertrauensagent, der auf Blockchain-Technologie basiert, könnte abhängig von der ausgewählten Blockchain-Technologie ebenfalls als Mikroservice implementiert werden. Ausführliche Informationen zur Architektur des Lizenz-Vertrauensagenten sind im nächsten Abschnitt 5.1.1 zu finden.

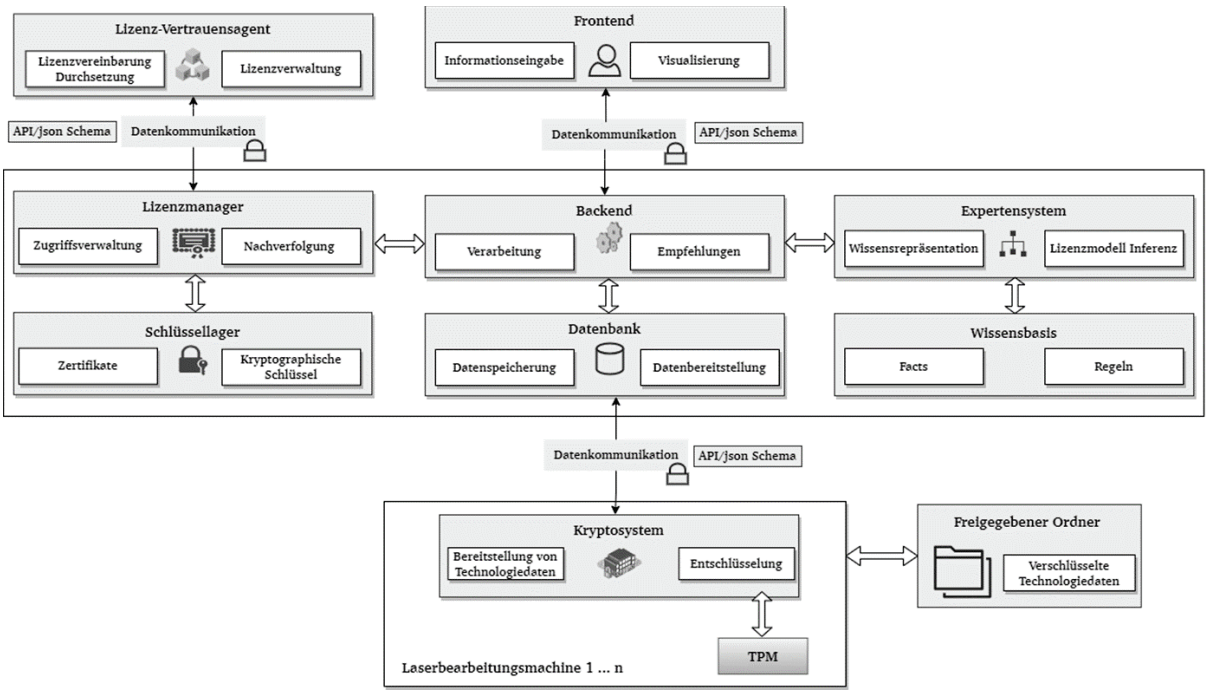

Abbildung 5-1: Überblick über die Architektur und den Aufbau der technischen Umsetzung des Gesamtsystems (eigene Darstellung)

Nun werden die dargestellten Komponenten des entwickelten Systems vorgestellt:

- Der **Lizenz-Vertrauensagent**: Diese Komponente trägt die Verantwortung für die Handhabung und Verwaltung der erworbenen Lizenzen, indem sie robuste Sicherheitsmaßnahmen gegen Manipulationen implementiert. Eine nachträgliche Änderung der Lizenzbedingungen ist in diesem Kontext strikt ausgeschlossen.

- Der **Lizenzmanager**: Diese Komponente fungiert als Vermittler zwischen dem Backend und dem Lizenz-Vertrauensagenten und ist für die Anwendung der API des Lizenz-Vertrauensagenten auf der Blockchain verantwortlich. Der Lizenzmanager stellt eine API-Schnittstelle für das Backend bereit, über die Lizenzinformationen abgerufen werden können, z. B. die Liste der erworbenen Lizenzen und deren Nutzungshistorie. Die Zugriffsanfragen auf eine bestimmte Lizenz werden auch darüber gesendet, um Entschlüsselungsinformationen zu erhalten. Der Lizenzmanager verwendet kryptographische Informationen, die in einer Keystore-Datenbank gespeichert werden, um eine sichere Verbindung zum Lizenz-Vertrauensagenten zu initialisieren und die Authentizität und Autorisierung des jeweiligen Nutzers sicherzustellen.

- Das **Kryptosystem**: Diese Komponente ist ein wesentlicher Bestandteil der Datensicherheit. Das Kryptosystem kommuniziert über eine vordefinierte API mit dem Backend, um bei Bedarf die Entschlüsselung von Technologiedaten zu ermöglichen und diese den entsprechenden Laserverarbeitungsmaschinen bereitzustellen. Dabei kommen verschiedene Instanzen des Kryptosystems zum Einsatz, wobei jede Instanz die Technologiedaten für eine spezifische Laserbearbeitungsmaschine bereitstellt. Die Backend-Anwendung sendet eine Anfrage zur Entschlüsselung der benötigten Technologiedaten zusammen mit den Lizenzinformationen und den verschlüsselten Entschlüsselungsdaten an das Kryptosystem. Zur Entschlüsselung dieser Daten greift das Kryptosystem auf ein *Trusted Platform Module* (TPM) zurück, in dem der private Schlüssel des Benutzers gespeichert ist. Anschließend holt das Kryptosystem die verschlüsselten Technologiedaten basierend auf ihrer Identifikationsnummer in den Lizenzinformationen ab, die sich in einem freigegebenen Ordner im privaten Netzwerk des Benutzers befinden. Schließlich entschlüsselt es die Technologiedaten und stellt sie der Laserbearbeitungsmaschine zur Verfügung.

- Das **wissensbasierte Assistenzsystem**: Diese Komponente nutzt die Wissensbasis und Inferenzregeln, um passende Lizenzmodelle zu ermitteln. Sie verarbeitet Nutzereingaben, die vom Backend bereitgestellt werden, konvertiert sie in Fakten, auf die die Inferenzregeln zugreifen können, und sendet diese an die Inferenzmaschine. Die Inferenzmaschine gibt dann die passenden Lizenzmodelle an das Backend zurück.

- Das **Backend**: Das Backend stellt den Kern des Systems dar und ist für die Verarbeitung und Vorbereitung der Daten sowie die Durchführung der erforderlichen Prozesse für die Auswahl von Laserbearbeitungsmaschinen und Lizenzmodellen verantwortlich. Das

Backend nimmt auch Anfragen von Nutzern über das Frontend entgegen, verarbeitet diese Anfragen und sendet die resultierenden Antworten und Ergebnisse an das Frontend zurück. Die Kommunikation zwischen dem Backend und den anderen Komponenten erfolgt über ein JSON-Schema. Die Speicherung und Bereitstellung der Daten wird in der Datenbank durchgeführt. Hierbei werden alle eingegebenen und anfallenden Daten in einer bereits vorher definierten Struktur abgespeichert.

- Das **Frontend**: Dies ist die Benutzeroberfläche, über die Daten eingegeben und die Ergebnisse der Datenverarbeitung visualisiert werden.

In den folgenden Kapiteln wird die Implementierung der dargestellten Komponenten beschrieben.

5.1.1 Implementierung des Lizenz-Vertrauensagenten

Der Lizenz-Vertrauensagent ist für die Verwaltung der erworbenen Lizenzen zuständig. Er wird auf der Basis von Blockchain-Technologie implementiert. Somit wird eine hohe Sicherheit gegen Manipulationen gewährleistet, da die Lizenzbedingungen nicht nachträglich geändert werden können. Zusätzlich müssen sämtliche Aktionen, die mit Lizenzen verbunden sind, transparent umgesetzt werden. Diese Maßnahmen sind notwendig, um die Sicherheitsanforderungen hinsichtlich Integrität, Authentizität, Autorisierung und Verfügbarkeit zu gewährleisten. Nun werden die verschiedenen Blockchain-Typen vorgestellt und die passende Art für die Implementierung wird gewählt. Danach werden die Einrichtung des Blockchain-Netzwerks sowie die Programmierung des Smart Contracts dargelegt.

Auswahl des passenden Blockchain-Netzwerk-Typs

Es gibt verschiedene Ausprägungen von Blockchain-Netzwerken, die jeweils eigene charakteristische Eigenschaften und Nutzungsszenarien aufweisen. Die am weitesten verbreiteten Typen sind öffentliche und private Blockchains.

Öffentliche Blockchains wie *Bitcoin* [167] und *Ethereum* [168] sind für alle zugänglich. Sie sind dezentralisiert, was bedeutet, dass es keine zentrale Autorität gibt, die das Netzwerk leitet. Transaktionen werden durch ein Netzwerk von Knotenpunkten validiert. Sobald sie bestätigt sind, werden sie dauerhaft in der Blockchain gespeichert. Öffentliche Blockchains zeichnen sich durch ihre dezentrale Struktur und die Anwendung kryptographischer Technologien zur

Sicherung von Transaktionen aus, was zu einer hohen Sicherheitsstufe führt. Allerdings sind sie aufgrund ihrer hohen Transaktionskosten und langen Bestätigungszeiten für viele Anwendungsbereiche nicht geeignet. Zudem gewährleisten sie aufgrund ihrer öffentlichen Natur keine absolute Privatsphäre, da alle Transaktionen für jeden im Netzwerk sichtbar sind.

Im Gegensatz dazu sind private Blockchains für den Einsatz innerhalb einer einzelnen Organisation konzipiert. Sie bieten einen besseren Datenschutz als öffentliche Blockchains, da sie zugangsbeschränkt sind, was bedeutet, dass der Zugang zum Netzwerk auf eine spezifische Gruppe von Teilnehmern begrenzt ist. Private Blockchains sind effizienter und kosteneffektiver als öffentliche Blockchains [169], da ausschließlich autorisierte Teilnehmer Transaktionen einsehen und validieren können. Im Vergleich zu öffentlichen Blockchains sind private Blockchains zumeist schneller und kostengünstiger, da sie nicht dieselbe umfangreiche Validierung durch ein großflächiges Netzwerk von Knoten benötigen. Allerdings sind sie daher auch weniger sicher als öffentliche Blockchains, da sie sich auf eine zentrale Autorität verlassen, um die Validierung von Transaktionen durchzuführen [90]. Die Auswahl des am besten geeigneten Blockchain-Typs für einen bestimmten Anwendungsfall hängt von verschiedenen Faktoren ab, darunter Sicherheitsanforderungen, Geschwindigkeit, Privatsphäre und die Anzahl der Teilnehmer. Öffentliche Blockchains sind zwar in Bezug auf die Sicherheit hochgradig zuverlässig, jedoch in ihrer Geschwindigkeit begrenzt und bieten nur einen eingeschränkten Datenschutz. Private Blockchains hingegen zeichnen sich durch ihre hohe Geschwindigkeit und einen verbesserten Datenschutz, sind aber in ihrer Sicherheit nicht so robust wie öffentliche Blockchains. Aus der Diskussion zu den Blockchain-Typen ergibt sich als Schlussfolgerung, dass für die prototypische Implementierung des entwickelten Systems eine private Blockchain gewählt wird.

Für die Datenkommunikation innerhalb die Blockchain werden spezielle Datenübertragungswege verwendet, die eine sichere und private Kommunikation zwischen ausgewählten Netzwerkmitgliedern ermöglichen [170]. Diese Wege sind besonders relevant, wenn es darum geht, sensible Informationen zu schützen. Beispielsweise werden für die Distributed-Ledger-Plattform *Hyperledger Fabric* sogenannte „Kanäle“ genutzt, um private Kommunikationspfade zwischen Netzwerkmitgliedern zu etablieren [171]. Jeder Kanal führt sein eigenes Ledger, das alle Transaktionen zwischen den Mitgliedern dieses spezifischen Kanals verzeichnet. Ähnlich werden in *Corda* „Flows“ genutzt. Flows sind so konzipiert, dass sie Privatsphäre und Vertraulichkeit garantieren und es den Netzwerkmitgliedern ermöglichen, sicher miteinander zu kommunizieren [172].

Für die Umsetzung des privaten Blockchain-Netzwerks in dieser Dissertation wird das *Hyperledger Fabric*-Framework eingesetzt. *Hyperledger Fabric* ist ein weit verbreitetes privates Open-Source-Blockchain-Framework, das die Entwicklung von verteilten Anwendungen ermöglicht. Zudem bietet es private Kommunikationswege „Kanäle“, die eine sichere und private Kommunikation zwischen bestimmten Netzwerkmitgliedern gewährleisten. Aufgrund dieser Eigenschaften eignet sich *Hyperledger Fabric* ideal für die Implementierung des Lizenz-Vertrauensagenten. Zunächst werden die erforderlichen Grundlagen für diese Implementierung beschrieben.

Grundlagen der Hyperledger Fabric

In einem Netzwerk, das auf *Hyperledger Fabric* basiert, kooperieren diverse Parteien, um ein gemeinschaftliches, manipulationssicheres Hauptbuch (engl. Ledger) mit Transaktionseinträgen zu erstellen und zu pflegen. Dieses Hauptbuch dient allen Teilnehmern als vertrauenswürdige Informationsquelle. Die zentralen Komponenten eines *Hyperledger Fabric*- Netzwerks werden im Folgenden vorgestellt:

- Peer-Knoten (engl. Peer Nodes): Diese Knoten sind für die Führung des Hauptbuchs und die Implementierung des Chaincodes (engl. Smart Contracts) verantwortlich. Sie speichern die Daten in der Blockchain, stellen den Konsens innerhalb des Netzwerks sicher und validieren die Transaktionen [173].

- Klienten (engl. Clients): Das sind Anwendungen, die mit dem Netzwerk interagieren. Sie dienen dazu, Transaktionen einzureichen, das Hauptbuch abzufragen und das Netzwerk über das *Fabric Gateway* zu verwalten [174].

- Der Membership Service Provider (MSP) ist für die Verwaltung der Teilnehmeridentitäten im Netzwerk verantwortlich. Zertifizierungsstellen stellen digitale Zertifikate *X.509* aus und legen sie im MSP ab, so dass eine Identität vom Netzwerk erkannt werden kann [175]. Innerhalb des MSP sind die Identitäten in einer bestimmten Ordnerstruktur auf dem Dateisystem jedes Knotens organisiert. Jeder Ordner enthält das notwendige kryptografische Material, wie beispielsweise die Schlüssel und Zertifikate, für eine spezifische Organisation oder Einheit innerhalb des Netzwerks. Die Abbildung 5-2 veranschaulicht die Ordnerstruktur einer MSP für eine Einheit im Netzwerk [176].

- Der Kanal (engl. Channel): Dies ist eine logische Partition eines Netzwerks. Sie ermöglicht verschiedenen Gruppen von Teilnehmern, miteinander zu interagieren, ohne dass sie ihre Daten mit dem gesamten Netzwerk teilen müssen [171].

- Der Auftraggeber (engl. Orderer): Diese Komponente ist verantwortlich für die Anordnung der Transaktionen auf eine deterministische Art und Weise. Sie stellt sicher, dass alle Knoten im Netzwerk über die Reihenfolge der Transaktionen im Einklang sind. Der Ordnungsdienst wird mit verschiedenen Algorithmen implementiert, einschließlich *Raft*, *Kafka* und *Solo* [177]. Dabei ist *Raft* die empfohlene Implementierung. Die anderen Implementierungen werden ab der Version 2.x nicht mehr empfohlen.

- Der Intelligente Vertrag (engl. Chaincode): Dies entspricht dem Smart Contract, der innerhalb des Netzwerks implementiert wird.

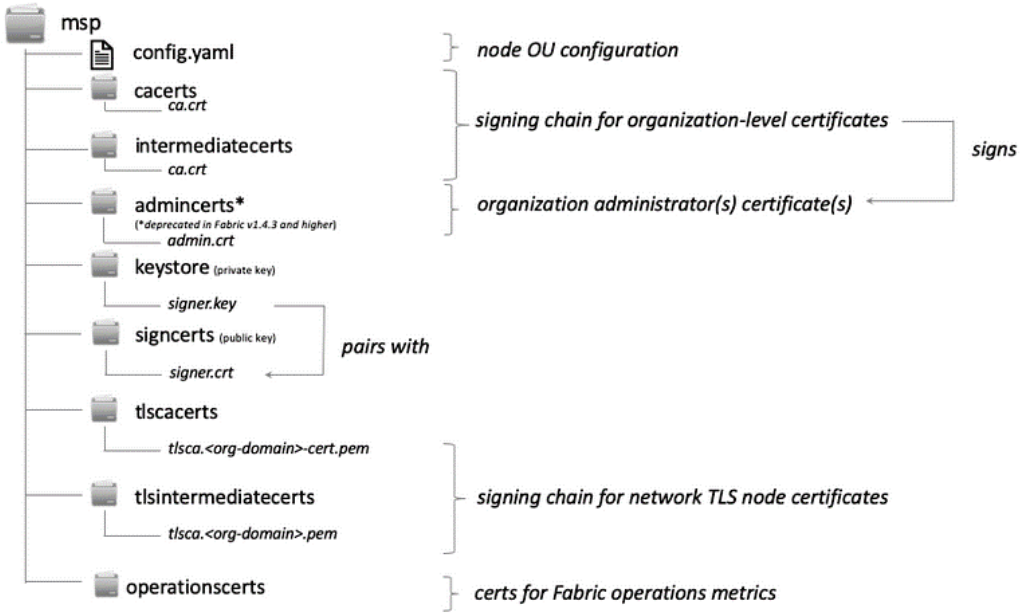

Abbildung 5-2: Verzeichnisstruktur eines MSPs (Quelle [176])

Eine Identität in *Hyperledger Fabric* repräsentiert ein digitales Zertifikat, das für die Authentifizierung und Autorisierung von Benutzern, Anwendungen, Peer-Knoten und anderen Einheiten innerhalb des Netzwerks verwendet wird [175]. Das Zertifikat beinhaltet den öffentlichen Schlüssel des Nutzers, der zur Überprüfung seiner Signatur auf Transaktionen herangezogen wird. Zudem enthält das Zertifikat die Identitätsinformationen des Nutzers, wie beispielsweise den Namen, die zugehörige Organisation und seine Rolle.

Wenn ein Nutzer einem *Hyperledger Fabric*-Netzwerk beitreten möchte, muss für ihn zunächst ein Zertifikat im MSP angelegt werden. Wenn der Nutzer eine Transaktion an das Netzwerk sendet, muss er die Transaktion zuvor mit seinem privaten Schlüssel signieren. Die Signatur wird dann vom Netzwerk mithilfe des abgelegten öffentlichen Schlüssels des Nutzers überprüft. Ist die Signatur gültig, wird die Transaktion vom Netzwerk bearbeitet. Dieser Authentifizierungsprozess zielt darauf ab, sicherzustellen, dass nur autorisierte Nutzer Zugang zum Netzwerk erhalten und Transaktionen durchführen können. Darüber hinaus soll der Prozess verhindern, dass sich nicht autorisierte Nutzer als autorisierte Nutzer ausgeben. Zum bessren Verständnis wird im Folgenden ein Beispiel für ein *Hyperledger Fabric*-Netzwerk gegeben.

Die Abbildung 5-3 illustriert ein exemplarisches Netzwerk in *Hyperledger Fabric*, an dem vier Organisationen (R0, R1, R2 und R3) teilnehmen. Innerhalb dieses Netzwerks gibt es zwei Kanäle (C1 und C2) für die Kommunikation zwischen den Teilnehmern. Die Knoten P1, P2 und P3, welche als „Peers“ bezeichnet werden, verwalten Kopien des Hauptbuchs und können Transaktionen ausführen.

In diesem Netzwerk gehört jeder Peer einer Organisation an und kann mehreren Kanälen beitreten. Im dargestellten Beispiel ist P1 von Organisation R1 Teilnehmer im Kanal C1, während P2 und P3 von den Organisationen R2 und R3 an beiden Kanälen, C1 und C2, teilnehmen. Jeder Peer erhält eine Kopie des Ledgers für jeden Kanal, zu dem er Zugang hat. Zum Beispiel verwaltet P1 eine Kopie von L1, während P2 und P3 Kopien von L1 und L2 halten, da sie Zugang zu den Kanälen C1 und C2 haben. Andererseits hat P1 keinen Zugriff auf L2 und ist sich der Existenz von Kanal C2 nicht bewusst. In ähnlicher Weise wird eine Kopie des intelligenten Vertrags (engl. Chaincode) auf den Peers jedes Kanals installiert, der an der Kommunikation teilhat. Im gegebenen Beispiel wird der Smart Contract S5, der zu Kanal C1 gehört, auf allen Peers (P1, P2 und P3) dieses Kanals installiert. Im Gegensatz dazu wird S6 nicht auf P1 installiert, da P1 kein Mitglied des Kanals C2 ist.

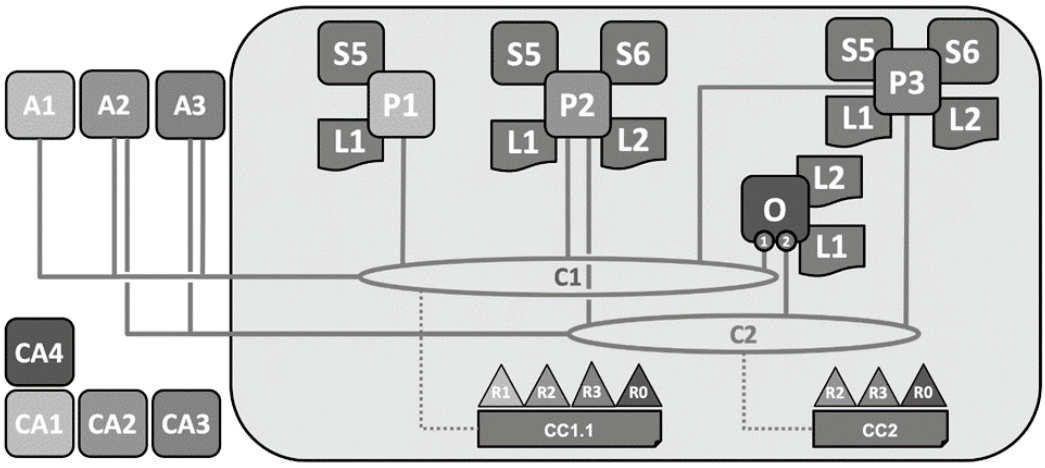

Abbildung 5-3: Ein Beispiel zum Hyperledger-Fabric-Netzwerk (Quelle [178])

Die Organisationseinheiten (OUs, Organizational Units) bei den Knoten der *Hyperledger Fabric* werden zur Definition der Rollen und Berechtigungen der Knoten genutzt [176]. Knoten sind die Einheiten, die am Konsensprotokoll teilnehmen und verschiedene Aufgaben erfüllen, wie beispielsweise die Bestätigung von Transaktionen, die Validierung von Blöcken und die Wartung des Ledgers. Eine Organisation kann in mehrere OUs unterteilt sein, wobei jede Einheit spezifische Verantwortlichkeiten hat.

Die Rollen werden durch den OU-Wert im *CommonName*-Attribut des *X509*-Zertifikats [179] bestimmt, welches die Identität eines Knotens innerhalb des Netzwerks repräsentiert. In der *Hyperledger Fabric* können Knoten basierend auf ihren Rollen und Verantwortlichkeiten innerhalb des Netzwerks in verschiedene Organisationseinheiten (OUs) klassifiziert werden. Es gibt vier vordefinierte OUs [176]:

- Peer OU: Diese OU umfasst alle Peer-Knoten im Netz. Diese Einheit enthält die Knoten, die Transaktionen überprüfen und die Blockchain pflegen. Diese Knoten sind an der Erstellung neuer Blöcke beteiligt und stellen sicher, dass die Transaktionen den Regeln des Netzwerks entsprechen.

- Orderer OU: Diese OU umfasst alle Orderer-Knoten im Netzwerk. Die Knoten in dieser Einheit repräsentieren die Endbenutzer, die Transaktionen einreichen und abfragen können. Sie initiieren Transaktionen und können das Ledger abfragen, haben jedoch keine Möglichkeit, neue Blöcke zu erstellen oder Transaktionen zu validieren

- Client OU: Diese OU beinhaltet alle Client-Knoten im Netzwerk. Client-Knoten sind verantwortlich für das Einreichen von Transaktionen und das Abfragen des Ledgers.

- Admin OU: Diese OU beinhaltet alle administrativen Knoten im Netzwerk. Administratoren haben die meisten Zugriffsrechte und können Aufgaben wie das Erstellen von Kanälen, das Hinzufügen oder Entfernen von Knoten und das Aktualisieren der Netzwerkkonfiguration durchführen.

Die OU-Rollen von Knoten werden in der *config.yaml* im MSP definiert. Im folgenden Beispiel wird eine Klient-Rolle definiert, indem ein Zertifikat unter Verwendung des CA-Zertifikats *ca.sampleorg-cert.pem* ausgestellt und das OU-Attribut als *OU=client* festgelegt wird.

In ähnlicher Weise können die Rollen anderer OUs, wie Peer, Orderer und Admin, in den entsprechenden Abschnitten der *config.yaml*-Datei konfiguriert werden, indem jeweils ein spezifisches CA-Zertifikat (Certificate Authority) verwendet und das passende OU-Attribut festgelegt wird. Jede Organisationseinheit (OU) verfügt über einen spezifischen Satz von Berechtigungen, die durch entsprechende Richtlinien festgelegt werden [180].

```
NodeOUs:
  Enable: true
  ClientOUIdentifier:
    Certificate: cacerts/ca.sampleorg-cert.pem
    OrganizationalUnitIdentifier: client
  PeerOUIdentifier:
    Certificate: cacerts/ca.sampleorg-cert.pem
    OrganizationalUnitIdentifier: peer
  AdminOUIdentifier:
    Certificate: cacerts/ca.sampleorg-cert.pem
    OrganizationalUnitIdentifier: admin
  OrdererOUIdentifier:
    Certificate: cacerts/ca.sampleorg-cert.pem
```

Prozedur 5-1: Beispiel für eine Klient-Rolle (Quelle [176])

Diese Berechtigungen bestimmen die Aktionen, die ein Knoten innerhalb des Netzwerks durchführen kann. Durch die Kategorisierung von Knoten in verschiedene OUs und die Zuweisung entsprechender Berechtigungen kann das Netzwerk gesichert und die Rollen und Verantwortlichkeiten der Knoten können klar definiert werden.

Einrichtung des Hyperledger-Fabric-Netzwerks für den Lizenz-Vertrauensagenten

Die Implementierung des Lizenz-Vertrauensagenten in einem privaten *Hyperledger Fabric*-Netzwerk setzt die Einrichtung mehrerer Organisationen voraus. Jede dieser Organisationen repräsentiert entweder einen Technologiedatenmarktplatz oder einen Anwender. Die Anforderungen für die Einrichtung des Netzwerks für den Lizenz-Vertrauensagenten werden in der Tabelle 5-1 dargestellt. Jede Organisation verfügt über einen eigenen Pool an Teilnehmern. Jeder Teilnehmer, kann durch einen oder mehrere Peers vertretet sein. Als Teilnehmer kommen Technologiedatenmarktplätze und Anwender in Frage. Diese Teilnehmer haben die Möglichkeit, diversen Kanälen beizutreten, was ihnen die Verwaltung ihrer Lizenzen über das Blockchain-Netzwerk ermöglicht. Für jedes Paar von Technologiedatenmarktplatz und Anwender wird ein separater Kanal eingerichtet, um ein hohes Maß an Sicherheit und Datenschutz zu gewährleisten. Dabei ist es für Anwender möglich, mehreren Kanälen beizutreten. Diese Konfiguration ermöglicht den Anwendern einen flexiblen und effizienten Erwerb von Lizenzen auf unterschiedlichen Technologiedatenmarktplätzen.

Tabelle 5-1: Anforderungen an die Implementierung des Lizenz-Vertrauensagenten

Nr.	Art	Bezeichnung
Anforderungen an die Implementierung des Netzwerks des Lizenz-Vertrauensagenten		
1	F	Das Netzwerk soll jedem Paar Technologiedatenmarktplatz/Anwender einen separateren Kanal bereitstellen.
2	F	Jeder Nutzer muss mindestens einen Peer haben, der seine Organisation im Netzwerk repräsentiert.
3	F	Technologiedatenmarktplätze sollen in der Lage sein, intelligente Verträge auf den entsprechenden Kanälen bereitzustellen.

4	F	Das Netzwerk soll dem Anwender den Zugriff auf die erworbenen Lizenzen ermöglichen.
5	F	Das Netzwerk soll dem Anwender ermöglichen, sich auf mehreren Kanälen gleichzeitig mit weiteren Technologiedatenmarktplätzen zu verbinden.

Den Kanälen können ausschließlich Peers von Technologiedatenmarktplätzen beitreten, auf denen Lizenzen erworben wurden. Technologiedatenmarktplätze tragen dabei die Verantwortung für die Bereitstellung der Lizenzen auf dem Ledger des entsprechenden Netzwerk-Kanals. Sobald die vertraglichen Bedingungen erfüllt wurden, soll es dem Anwender möglich sein, auf die erworbenen Lizenzen zuzugreifen.

Die Abbildung 5-4 veranschaulicht die Implementierung eines *Hyperledger Fabric*-Netzwerks mit zwei Organisationen A und X, wobei A einen Technologiedatenmarktplatz und X einen Anwender repräsentiert.

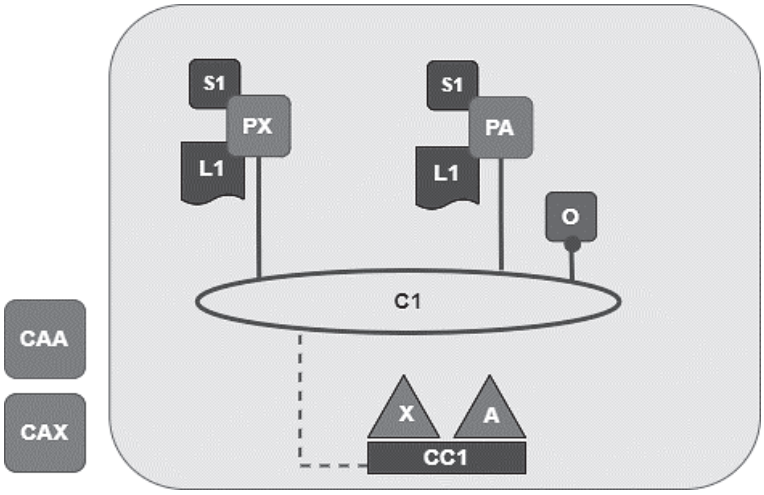

Abbildung 5-4: Hyperledger-Fabric-Netzwerk mit einem Kanal für einen Technologiedatenmarktplatz A und einen Anwender X (eigene Darstellung)

Zur Sicherstellung einer sicheren und privaten Kommunikation zwischen beiden Teilnehmern wird ein spezifischer Kommunikationskanal, C1, eingerichtet. Innerhalb des Kanals C1 existieren zwei Peers: PA, der zum Technologiedatenmarktplatz A gehört, und PX, der zum Anwender X gehört. Zudem gibt es zwei Zertifizierungsstellen, CAX und CAA, die jeweils für X

und A zuständig sind. Das Ledger L1, das zu Kanal C1 gehört, dient zur Aufzeichnung aller durchgeführten Transaktionen. Es ist nur für die innerhalb des Kanals definierten Teilnehmer, den Technologiedatenmarktplatz A und den Anwender X, einsehbar und zugänglich. Dies stellt eine zusätzliche Sicherheits- und Datenschutzebene dar, da nur autorisierte Teilnehmer Zugang zu den Transaktionsdaten haben.

Die Konfiguration CC1 stellt die Struktur und die Parameter für den Kanal C1 dar. Sie definiert einen Kanal namens C1 mit zwei Organisationen, A und X, als Mitgliedern. Darüber hinaus werden in dieser Konfiguration die Richtlinien festgelegt, die bestimmen, welche Rollen jede Organisation innerhalb des Kanals einnehmen kann.

Das MSP dient hierbei als Identitätsverzeichnis. Jede Organisation bekommt eine MSP-ID und einen Pfad zu ihrem MSP-Ordner, der alle wichtigen Identitätsinformationen enthält. Dazu gehören die in der Datei *config.yaml* definierten Rollen sowie die Root-Zertifikate der Zertifizierungsstelle der Organisation, die zur Erstellung der Zertifikate für alle Identitäten innerhalb der Organisation verwendet werden.

Des Weiteren wird die Auftraggeber-Organisation O (engl. Orderer) in dieser Konfiguration definiert. Diese Organisation fungiert als Auftraggeber für die Durchführung von Transaktionen und ist für die Verwaltung aller zugehörigen Knoten im Netzwerk verantwortlich. Der Auftraggeber gewährleistet die konsistente und korrekte Reihenfolge von Transaktionen im Netzwerk, um sicherzustellen, dass alle Transaktionen korrekt und einheitlich verarbeitet werden. Jede Organisation legt eine Reihe von Richtlinien fest, die bestimmen, welche Akteure (Knoten, Benutzer, usw.) Transaktionen in einem Channel lesen, schreiben und bestätigen können. Diese Richtlinien werden durch die Felder *Type* und *Rule* definiert. Diese Felder definieren die Berechtigungen und Rollen der Akteure innerhalb einer Organisation. *Type* könnte eine Rolle wie *Admin*, *Peer* oder *Client* spezifizieren. *Rule* definiert die spezifischen Aktionen, die im Rahmen einer Rolle ausgeführt werden können. Die Rolle *Client* wird in der Regel einer Anwendung oder einem Benutzer zugewiesen, der sich über einen Peer-Knoten mit dem Netzwerk verbindet und Transaktionen initiieren oder Daten abfragen kann.

Für die Organisationen des Technologiedatenmarktplatzes A und des Anwenders X sind die folgenden Richtlinien definiert:

1. Jeder Anwender mit der Rolle „Admin“, „Peer“ oder „Client“ kann Transaktionen lesen.

2. Jeder Anwender mit der Rolle „Admin“ oder „Client“ kann Transaktionen schreiben.

3. Nur Anwender mit der Rolle „Admin“ können die Organisation verwalten.

4. Nur Anwender mit der Rolle „Peer“ können Transaktionen genehmigen.

Jede Anwendung verfügt über einen eigenen Satz von Richtlinien, die festlegen, wer Transaktionen im Kanal lesen, schreiben, bestätigen und verwalten darf. Die Anwendung ist eine Software, die sich außerhalb des Netzwerks befindet, sich mit dem Netzwerk über einen Peer verbindet und zu einer Organisation gehört. Ein Beispiel für eine Anwendung im entwickelten Konzept ist der Lizenzmanager.

Jeder Benutzer verfügt innerhalb einer Organisation über eine Identität, die zur Authentifizierung der Anwendung im Netzwerk und zum Signieren von Transaktionen verwendet wird. Weiterhin gelten folgende Regelungen:

5. Jeder Benutzer kann die Transaktionen in seinen Kanälen lesen.

6. Jeder Benutzer kann Transaktionen in seinen Kanälen schreiben.

7. Eine Mehrheit der Organisationen im Netzwerk muss eine Transaktion genehmigen, bevor sie verwaltet werden kann.

8. Jede Änderung der Chaincode-Definition oder der zugehörigen Richtlinien muss von der Mehrheit der Organisationen im Netzwerk genehmigt werden.

9. Eine Transaktion muss von der Mehrheit der Organisationen im Netzwerk genehmigt werden, bevor sie in das Ledger übertragen wird.

Diese Regelungen sorgen dafür, dass das Netzwerk ordnungsgemäß funktioniert, und sie gewährleisten einen hohen Grad an Sicherheit und Vertrauen zwischen den Teilnehmern. Die Mehrheitsbedingung in den Richtlinien bedeutet, dass sowohl der Anwender als auch der Technologiedatenmarktplatz eine Transaktion unterzeichnen muss, bevor sie im Netzwerk genehmigt wird. Wenn beispielsweise ein intelligenter Vertrag im Netzwerk bereitgestellt wird, muss er sowohl vom Technologiedatenmarktplatze als auch vom Anwender mit ihren privaten Schlüsseln unterzeichnet werden, bevor er genehmigt werden kann.

Ein weiteres Beispiel ist, wenn ein Benutzer eine Anfrage für den Lizenzzugang über den Lizenzmanager sendet. In diesem Fall signiert der Lizenzmanager als Identität des Anwenders

diese Anfrage und übermittelt sie an den Lizenz-Vertrauensagenten. Der Smart Contract verarbeitet die Anfrage und erzeugt die entsprechende Transaktion als Antwort. Die Antworttransaktion wird automatisch sowohl vom Technologiedatenmarktplatz als auch vom Anwender von ihren Peer-Knoten signiert, da sie Kopien des Ledgers besitzen, bevor die Transaktion im Ledger jedes Peers gespeichert wird. Dies geschieht automatisch, da die Transaktion durch den intelligenten Vertrag erzeugt wird, der von beiden Parteien unterzeichnet wurde. PA oder PX können auch anderen Kanälen mit anderen Peers beitreten. Jedoch ist es wichtig, dass jeder Kanal nur einen Technologiedatenmarktplatz mit einem Anwender verbindet.

Im Folgenden werden weitere zwei beispielhafte Netzwerke für mehrere Teilnehmer dargestellt. Auf der Abbildung 5-5 wird ein Netzwerk für einen Technologiedatenmarktplatz A und zwei Anwender X und Y dargestellt. Hierbei kommuniziert der Peer PA mit dem Peer PX über C1 und mit PY über C2. Auf der Abbildung 5-6 wird ein Netzwerk für zwei Technologiedatenmarktplätze A und B und einen Anwender X dargestellt. Hierbei kommuniziert der Peer PX mit dem Peer PA über C1 und mit PB über C2.

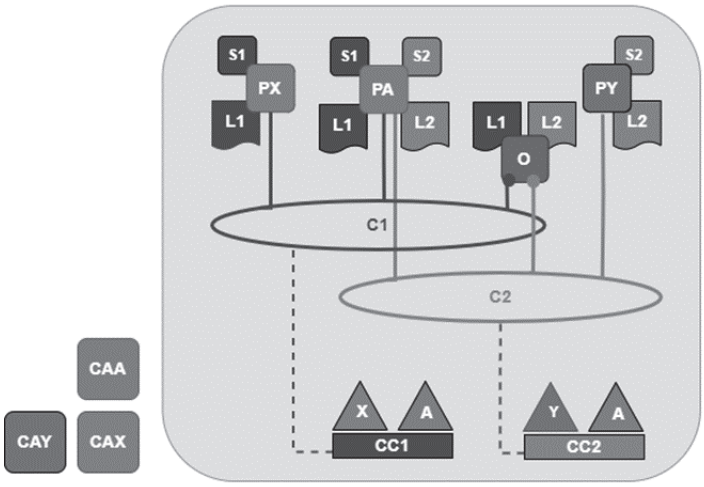

Abbildung 5-5: Hyperledger-Fabric-Netzwerk mit zwei Kanälen für einen Technologiedatenmarktplatz A und zwei Anwender: X und Y (eigene Darstellung)

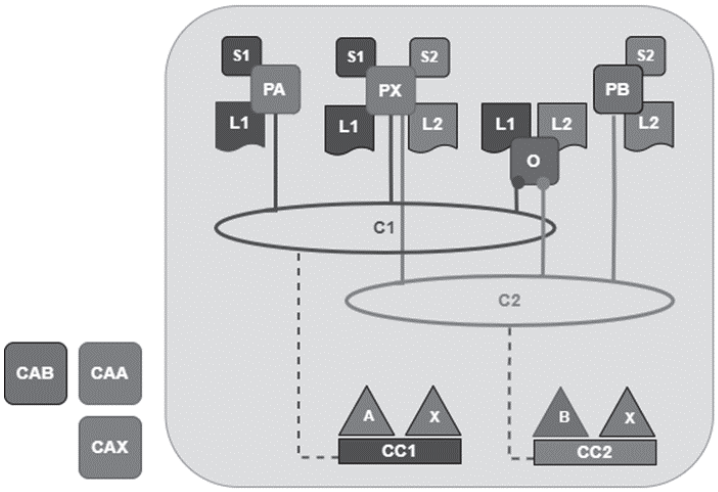

Abbildung 5-6: Hyperledger-Fabric-Netzwerk mit zwei Kanälen für zwei Technologiedatenmarktplätze A und B und einen Anwender X (eigene Darstellung)

Im Folgenden werden die erforderlichen Schritte zur Einrichtung des *Hyperledger Fabric*-Netzwerks für die prototypische Implementierung des Lizenz-Vertrauensagenten für einen Anwender und einen Technologiedatenmarktplatz (wie auf der Abbildung 5-4 dargestellt) erläutert. Hierbei wird ein Kanal zwischen einem Technologiedatenmarktplatz A und einem Anwender X erstellt und konfiguriert. Eine detaillierte Anleitung ist in Anhang 1 zu finden.

1. Installation der *Hyperledger Fabric*: Zunächst muss das *Hyperledger Fabric* installiert werden. Hierfür werden die Anweisungen in der Dokumentation [181] verfolgt.

2. Erstellen der Konfigurationsdateien: Als nächstes müssen die Konfigurationsdateien für das Netzwerk angelegt werden. Die Konfigurationsdateien definieren die Netzwerktopologie, die Organisationen und die Kanäle. Hier folgt eine Liste dieser eingerichteten Konfigurationsdateien:

   - *crypto-config.yaml*: Definiert das kryptografische Material für das Netzwerk, einschließlich der Organisationen, Peers und Auftraggeber.

   - *configtx.yaml*: Definiert die Konfiguration für die Kanäle, einschließlich der Organisationen, die an jedem Kanal teilnehmen.

   - *docker-compose.yaml*: Definiert die Docker-Container, auf denen die Netzwerkkomponenten (Peers, Orderer und Command Line Interface (CLI)) ausgeführt werden.

3. Generieren der kryptografischen Materialien: Nach der Erstellung der Konfigurationsdateien, müssen die kryptografischen Materialien für das Netzwerk mit dem von der *Hyperledger Fabric* bereitgestellten Werkzeug *cryptogen* [182] generiert werden. Dieses Werkzeug liest die Datei *crypto-config.yaml* und generiert die erforderlichen Zertifikate und Schlüssel für jede Organisation, jeden Peer und jeden Auftraggeber. In der Praxis wird aber empfohlen, dass jede Organisation ihre eigenen Zertifikate und privaten Schlüssel von einer bekannten und vertrauenswürdigen Zertifizierungsstelle bezieht.

4. Generieren der Netzwerk-Artefakte: Nach dem Generieren der kryptografischen Materialien müssen die Netzwerkartefakte mit dem von der *Hyperledger Fabric* bereitgestellten Werkzeug *configtxgen* [182] erzeugt werden. Dieses Werkzeug liest die Datei *configtx.yaml* und generiert die notwendigen Konfigurationsdateien für das Netzwerk.

5. Starten des Netzwerks: Nachdem die Netzwerk-Artefakte erstellt wurden, kann nun das Netzwerk mit dem Befehl *docker-compose* gestartet werden. Dieser Befehl liest die Datei *docker-compose.yaml* und startet die Docker-Container für die Peers, die Auftraggeber und die CLI.

6. Erstellen des Kanals: Nach dem Starten des Netzwerks können die Kanäle mithilfe des CLI-Containers erstellt werden. Im konkreten Beispiel wird einen Kanal für das Teilnehmerpaar: Technologiedatenmarktplatz A und Anwender X erstellt.

7. Verbinden von Peers mit dem Kanal: Nach dem Erstellen des Kanals können die Peers PA und PX mithilfe des CLI-Containers des Kanals beitreten. Hierfür wird den Befehl *peer channel join* verwendet.

Nun ist das *Hyperledger Fabric*-Netzwerk, das die festgelegten Anforderungen erfüllt, für die Implementierung des Lizenz-Vertrauensagenten eingerichtet. Jede Organisation wird durch einen Peer im Netzwerk vertretet. Für jedes Teilnehmerpaar wird ein privater Kanal zur Kommunikation erstellt.

Programmierung von Smart Contracts

Der intelligente Vertrag wird schließlich auf den Peers des Herstellers Technologiedatenmarktplatz A und des Anwenders X installiert und auf dem Ledger des bestehenden Kanals (A-X) gespeichert. Das bedeutet, dass ausschließlich A und X, die diesem Kanal beitreten können, auf den Vertrag zugreifen können. Die intelligenten Verträge können

alle im Konzeptkapitel genannten Lizenzmodelle abbilden. Wie bereits im Konzept erwähnt, werden vier repräsentative Lizenzmodelle im Rahmen der Implementierung berücksichtigt: *Einzelplatz, hardwarebasiert, zeitbegrenzt*; *Einzelplatz, hardwarebasiert, nutzungsbegrenzt*; *Mehrplatz, nutzerbasiert, zeitbegrenzt*; *Mehrplatz, nutzerbasiert, nutzungsbegrenzt*. Für jedes Lizenzmodell wird ein Smart Contract erstellt, unter Berücksichtigung des im Unterkapitel 4.5 vorgestellten Datenmodells. Für die Implementierung wurde die Programmiersprache *NodeJS* verwendet. Über die *Hyperledger Fabric* Contract API wurden Smart Contracts auf der Blockchain bereitgestellt [178]. In der Prozedur 5-2 wird ein Auszug aus dem programmierten Smart Contract dargestellt. Das veranschaulicht, wie der Smart-Contract-Code für eine Anwenderanfrage für den Zugriff auf eine Lizenz mit begrenzter Nutzungsanzahl lautet. In diesem Zusammenhang erfolgen die Authentifizierung des Anwenders und die Überprüfung der Lizenzgültigkeit, bevor die Entschlüsselungsinformationen geschickt werden.

```
    }

    // ReadLicense returns the license stored with given id.
    @Transaction(false)
    public async ReadLicense(ctx: Context, id: string): Promise<string> {
        if (id.endsWith('_ACTIVATION')) {
            throw new Error(`The license ${id} does not exist`);
        }
        // get the license from chaincode state
        const licenseJSON = await ctx.stub.getState(id);
        if (!licenseJSON || licenseJSON.length === 0) {
            throw new Error(`The license ${id} does not exist`);
        }
        return licenseJSON.toString();
    }

    // get the decryption information from the activation transaction
    private async ReadDecryptionInformation(ctx: Context, id: string): Promise<string> {
        const activationTransactionID = `${id}_ACTIVATION`;
        const activationTransactionJSON = await ctx.stub.getState(activationTransactionID);
        if (!activationTransactionJSON || activationTransactionJSON.length === 0) {
            throw new Error(`The decryption information ${id} does not exist`);
        }

        const license: LicenseModel = JSON.parse(activationTransactionJSON.toString());
        return license.DecryptionInformation;
    }

    // VerifyUser verifies that a user is authorized to use a license
    private VerifyUser(license: LicenseModel, user: string): boolean {
        return license.User === user;
    }

    // ValidateLicense verifies that a license is still active and valid
    // if the numUsages is equal to -1 then not check the number of usages.
    // if the expiration is empty string then not check the expiration date.
    private ValidateLicense(license: LicenseModel): boolean {
        let valid = true;
        // check expiration date
        if (license.Expiration !== '') {
            const now = new Date();
            const expirationDate = new Date(license.Expiration);
            if (now > expirationDate) {
                valid = false;
            }
```

```
        }
        // check number of usages
        if (license.NumUsages === 0) {
            valid = false;
        }
        return valid;
    }

    // AccessLicense checks if the license is still valid and
    // returns it with the decryption information after updating maxUsage
    @Transaction()
    public async AccessLicense(ctx: Context, id: string, user: string): Promise<string> {
        const licenseJSON = await this.ReadLicense(ctx, id);
        const license: LicenseModel = JSON.parse(licenseJSON);

        // check if it is active
        if (!license.Active) {
            throw new Error(`The license ${id} is not active`);
        }

        // verify user
        const VerifyUser = this.VerifyUser(license, user);
        if (!VerifyUser) {
            throw new Error(`The user ${user} is unauthorized to access the license ${id}`);
        }

        // validate license
        const isValid = this.ValidateLicense(license);
        if (!isValid) {
            // deactivate if it is invalid
            await this.DeactivateLicense(ctx, id)
            throw new Error(`The license ${id} is expired`);
        }

        // update NumUsages only if it is positive which means that the license is limited by
number of usage
        const updatedLicense = license;
        if (license.NumUsages > 0) {
            updatedLicense.NumUsages = license.NumUsages - 1;
            // the data is inserted in alphabetic order using 'json-stringify-deterministic' and
'sort-keys-recursive'
            await ctx.stub.putState(id,
Buffer.from(stringify(sortKeysRecursive(updatedLicense))));
        }

        // return the license with the decryption information
        updatedLicense.DecryptionInformation = await this.ReadDecryptionInformation(ctx, id);
        return stringify(sortKeysRecursive(updatedLicense));
    }
```

Prozedur 5-2: Auszug aus dem programmierten Smart Contract

Eine detaillierte Beschreibung der Programmierung und Installierung von intelligenten Verträgen ist in Anhang 2 zu finden.

5.1.2 Implementierung des Lizenzmanagers

Der Lizenzmanager ist für die Handhabung der vom Nutzer erworbenen Lizenzen verantwortlich. Einer seiner wichtigsten Aufgaben besteht darin, Lizenzen und deren Informationen vom Lizenz-Vertrauensagenten abzurufen. Seine Implementierung umfasst

Funktionen zum Erhalt von wichtigen Informationen über alle Lizenzen, die der Nutzer von verschiedenen Anbietern erworben hat. Dazu gehört auch das Abrufen der Historie einer spezifischen Lizenz oder aller erworbenen Lizenzen. Darüber hinaus übermittelt der Lizenzmanager Zugriffsanfragen für eine bestimmte Lizenz mit einer spezifischen Kombination aus Kanalname und Smart-Contract-ID an Lizenz-Vertrauensagenten. Für die Implementierung des Lizenzmanagers wurde die Programmiersprache *JavaScript* verwendet, ergänzt durch *TypeScript* für die Definition von Typen und Funktionen. Die Abbildung 5-9 veranschaulicht, wie eine sichere Verbindung zum Blockchain-Netzwerk aufgebaut und aufrechterhalten wird.

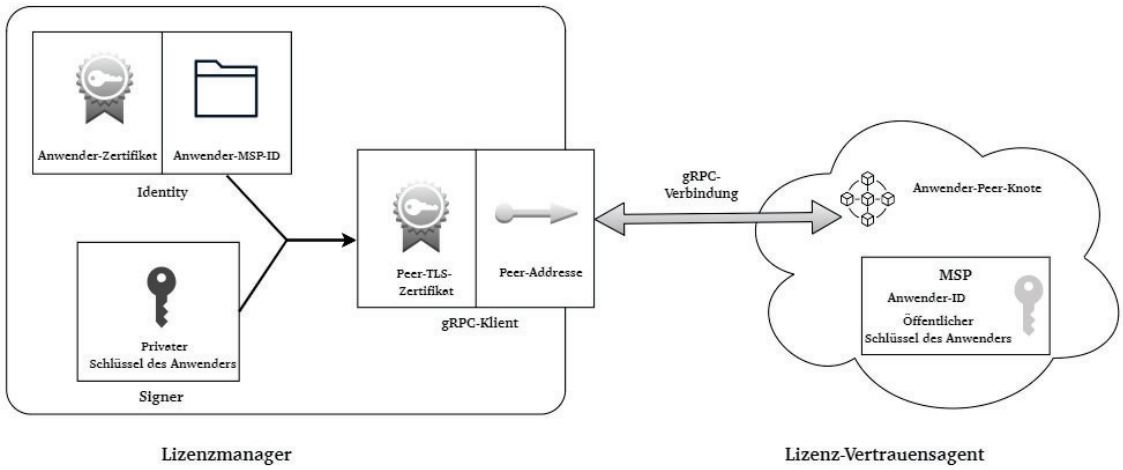

Abbildung 5-7: Sichere Verbindung zum Anwenders Knote auf dem Lizenz-Vertrauensagent im Netzwerk (eigene Darstellung)

Der Lizenzmanager nutzt das Zertifikat und den privaten Schlüssel des Benutzers für die Authentifizierung und Autorisierung beim Lizenz-Vertrauensagenten. Über die Schnittstelle *FabricIdentity* werden der Pfad zur Zertifikatsdatei des Nutzers und seine *MSP-ID* definiert, die auf die zugehörige MSP in der *Hyperledger Fabric* verweist, wo bereits Identitäten des Nutzers gespeichert sind. Mit der Funktion *newIdentity* wird ein neues Identitätsobjekt für den Nutzer erzeugt, indem sein Zertifikat und seine MSP-ID verwendet werden. Die Funktion *newSigner* erstellt ein neues Signatur-Objekt mit dem privaten Schlüssel des Nutzers. Beide Objekte, *Identity* und *Signer*, werden zusammen genutzt, um Transaktionen zu signieren und den Nutzer bei Interaktionen mit dem *Fabric*-Netzwerk zu authentifizieren. Darüber hinaus nutzt der Lizenzmanager das Peer-TLS-Zertifikat für eine sichere Kommunikation zwischen dem Lizenzmanager und dem Peer-Knoten des Nutzers im *Fabric*-Netzwerk. Der Lizenzmanager stellt eine Verbindung zum Peer-Knoten des Benutzers im *Hyperledger Fabric*-Netzwerk her, und zwar über das *Fabric Gateway* [174] mittels des *gRPC*-Protokolls [183]. Dies dient dazu, Zugriff auf

die im Ledger gespeicherten Lizenzen zu erhalten. Das *Fabric Gateway* ist ein Modul des *Fabric SDK* und bietet eine sichere und einfache Methode für Klienten-Anwendungen, um mit einem Fabric-Netzwerk zu kommunizieren. Es erleichtert die effiziente Kommunikation zwischen Klienten- und Serveranwendungen, und das über verschiedene Programmiersprachen und Plattformen hinweg.

Die Schnittstelle *PeerEndPoint* definiert die Peer-Adresse und den Pfad zur Datei des Peer-TLS-Zertifikats. Die Funktion *newClient* erzeugt einen neuen *gRPC*-Client, der eine Verbindung zum *Fabric*-Netzwerk herstellt und hierbei Informationen zum Endpunkt und zum Peer-TLS-Zertifikat verwendet, wie in Abbildung 5-7 dargestellt. Nachdem eine sichere Verbindung zum Peer-Knoten des Nutzers hergestellt wurde, kann der Lizenzmanager Anfragen an den Lizenz-Vertrauensagenten senden und Transaktionen übermitteln. Die an den Peer-Knoten des Nutzers gerichteten Anfragen sollten den Namen des Kanals und des Smart Contracts enthalten, mit dem interagiert werden soll. Dies ist erforderlich, weil der Peer-Knoten des Nutzers gleichzeitig mehreren Kanälen mit Peers unterschiedlicher Technologiedatenmarktplätze beitreten kann. Die Schnittstelle *SmartContractConfig* beinhaltet den Namen des Kanals und des Smart Contracts.

Die folgenden Funktionen wurden umgesetzt, um Funktionalitäten des Lizenzmanager bereitzustellen:

- Die Funktion *getListByChannel* dient dazu, alle Lizenzen für eine bestimmte Kombination aus festgelegtem Kanal und Smart Contract abzurufen.

- Die Funktion *GetAllLicenses* holt alle vom Anwender erworbenen Lizenzen von verschiedenen Marktplätzen. Sie durchläuft alle beigetretenen Kanäle und verwendet die Funktion *getListByChannel*, um alle Lizenzen für jede spezifische Kombination aus Kanal und Smart Contract abzurufen.

- Die Funktion *GetLicenseHistory* liefert die Historie einer bestimmten Lizenz inklusive der Zeitstempel.

- Die Funktion *AccessLicense* dient dazu, eine Transaktion zu senden, die den Zugriff auf eine spezifische Lizenz ermöglicht. Darüber hinaus werden auf diesem Weg Lizenzinformationen zusammen mit den verschlüsselten Entschlüsselungsinformationen bezogen.

Der Lizenzmanager bietet ein *RESTful* API (engl. Application Programming Interface) an, durch das die Backend-Applikation Zugriff auf seine Funktionen erhält. Um einen sicheren Zugriff zu gewährleisten, implementiert das API eine Basisauthentifizierung eingehender Anfragen. Das API akzeptiert nur Anfragen von einem bestimmten Rechner, um sicherzustellen, dass ausschließlich die Backend-Applikation mit ihm kommunizieren kann. Das API stellt vier Endpunkte bereit, durch die die Backend-Applikation auf die Funktionen des Lizenzmanagers zugreifen kann:

- Der Endpunkt */api/licenses* liefert alle Lizenzen für alle Kanäle und intelligenten Verträge.

- Der Endpunkt */api/licenses/:id* liefert Lizenzinformationen für eine spezifische Lizenz, basierend auf ihrer ID.

- Der Endpunkt */api/licenses/:id/history* liefert den Lizenzverlauf für eine spezifische Lizenz, basierend auf ihrer ID.

- Der Endpunkt */api/licenses/:id* reicht eine Transaktion ein, um Zugriff auf eine spezifische Lizenz zu beantragen.

Dank der Basisauthentifizierung können nur autorisierte Anfragen an das API des Lizenzmanagers gesendet werden, wodurch die Sicherheit sensibler Daten gewährleistet wird. Darüber hinaus bietet die Beschränkung, nur Anfragen vom gleichen Rechner zu akzeptieren, eine zusätzliche Sicherheitsebene, die unberechtigte Zugriffe von externen Quellen verhindert.

Sequenzdiagramm zur Funktion AccessLicense

Der Prozess des Zugriffs auf einer bestimmten Lizenz wird auf der Abbildung 5-8 als Sequenzdiagramm dargestellt. Zunächst beantragt ein bereits autorisiertes Konto den Zugriff auf die erworbenen Technologiedaten. Das autorisierte Konto öffnet dafür die Lizenzzugriffsseite, die den Zugriff auf die Technologiedaten über zugehörige erworbene Lizenz ermöglicht.

Auf der Seite für den Lizenzzugriff wählt das autorisierte Konto die Laserbearbeitungsmaschine(n) aus, auf der/denen die Technologiedaten verwendet werden sollen, und gibt sein Passwort für das Konto ein. Diese Informationen sind notwendig, damit

das System die Identität des Kontos überprüfen kann. Nach der Auswahl der Lasermaschine(n) und der Eingabe des Passworts klickt das autorisierte Konto auf die Schaltfläche *Verifizierungscode abrufen*. Dies löst eine Reihe von Interaktionen zwischen der Seite für den Lizenzzugriff und der Backend-Applikation aus.

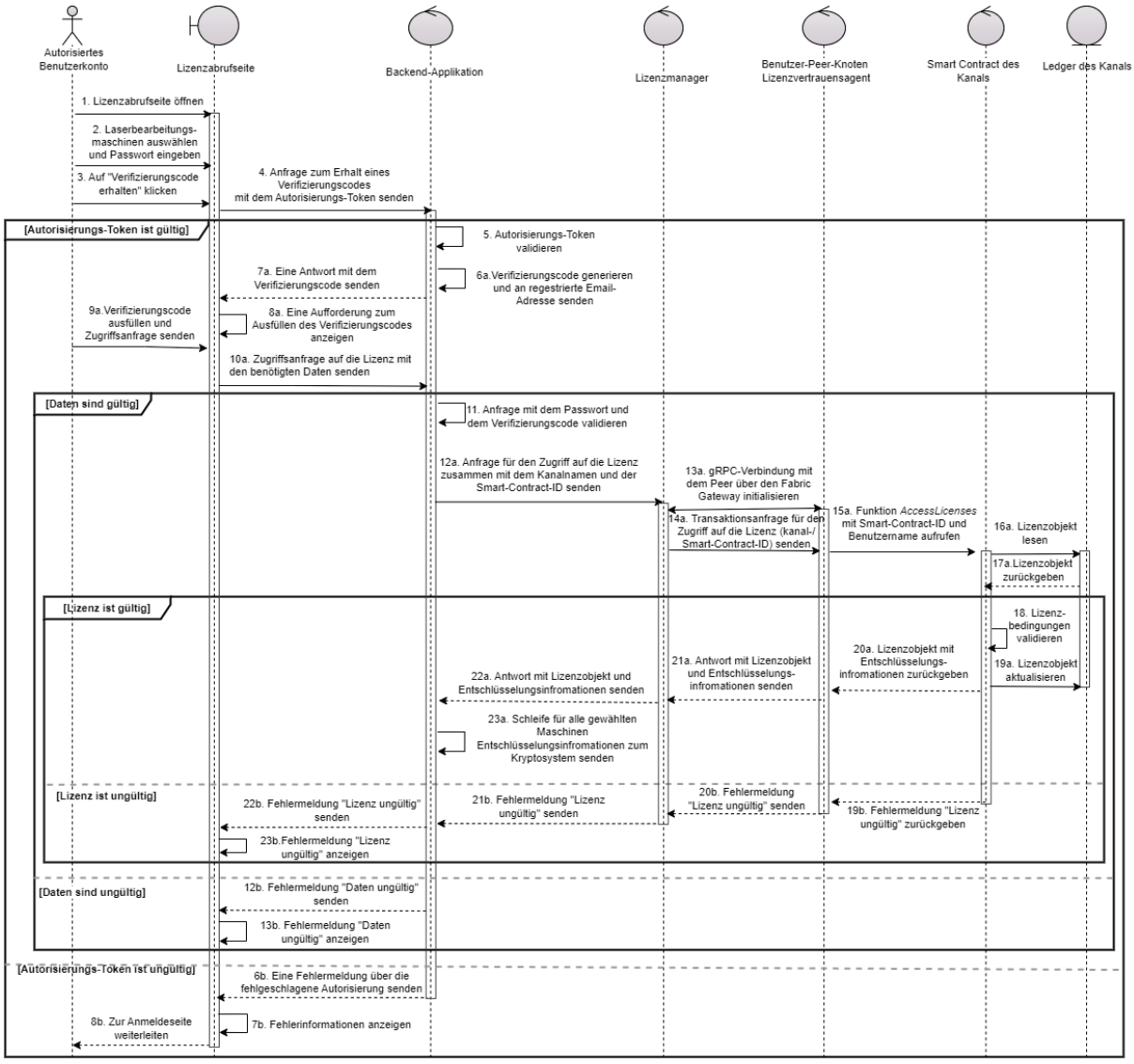

Abbildung 5-8 Sequenzdiagramm für den Zugriff auf einer Lizenz (eigene Darstellung)

Die Seite für den Lizenzzugriff sendet eine Anfrage an die Backend-Applikation, um einen Verifizierungscode zu erhalten. Diese Anfrage enthält das Autorisierungs-Token. Nach Erhalt der Anforderung von der Seite für den Lizenzzugriff validiert die Backend-Applikation das Autorisierungs-Token, um sicherzustellen, dass es gültig ist und dem autorisierten Konto entspricht, das den Zugriff auf die Technologiedaten anfordert. Ist das Autorisierungs-Token ungültig, sendet die Backend-Applikation eine Fehlermeldung an die entsprechende Seite für

den Lizenzzugriff. Die Seite für den Lizenzzugriff zeigt Fehlerinformationen an und leitet den Nutzer auf die Anmeldeseite weiter. Ist das Autorisierungs-Token gültig, generiert die Backend-Applikation einen Verifizierungscode und sendet ihn an die E-Mail-Adresse des autorisierten Kontos. Dieser Verifizierungscode dient als zusätzliche Sicherheitsmaßnahme, um zu garantieren, dass nur autorisierte Identitäten über die erworbene Lizenz auf die Technologiedaten zugreifen können.

Die Backend-Applikation sendet eine Antwort an die Seite für den Lizenzzugriff, um den Verifizierungscode vom autorisierten Konto anzufordern. Die Seite für den Lizenzzugriff zeigt eine Benachrichtigung für das autorisierte Konto an und fordert es auf, in seiner E-Mail nach dem Verifizierungscode zu suchen. Der Berechtigte ruft den Verifizierungscode aus seiner E-Mail ab und gibt ihn auf der Seite für den Lizenzzugriff ein, um einen Zugriffsantrag zu stellen. Diese Zugangsanfrage enthält den Verifizierungscode sowie alle anderen erforderlichen Daten, wie z. B. die Lasermaschine(n) und das Lizenz-Token.

Der Lizenzmanager initialisiert eine *gRPC*-Verbindung mit dem Peer-Knoten des Anwenders im Lizenz-Vertrauensagenten über das *Fabric Gateway*. Der Lizenzmanager sendet eine Transaktionsanforderung, um auf die Lizenz des gegebenen Channels/Smart Contracts zuzugreifen. Der Peer-Knoten des Anwenders im Lizenz-Vertrauensagenten ruft die Funktion *AccessLicense* aus dem Smart Contract auf und gibt dabei die Lizenz-ID, den Anwendernamen und die anderen in den Nutzungsbedingungen der Lizenz geforderten Daten an. Der Smart Contract des Kanals liest das Lizenzobjekt aus dem Ledger des Kanals und validiert die Lizenznutzungsbedingungen auf der Grundlage der in der Anfrage angegebenen Daten. Ist die Lizenz gültig, aktualisiert der intelligente Vertrag des Kanals das Lizenzobjekt im Ledger des Kanals, um alle Attribute zu ändern, die beim Zugriff auf die Lizenz geändert werden sollten (z. B. die Anzahl der Nutzungen), und zeichnet die Zugriffsanforderung auf. Der Smart Contract des Kanals sendet dann das aktualisierte Lizenzobjekt zusammen mit den verschlüsselten Entschlüsselungsinformationen an den Peer-Knoten des Anwenders zurück.

Der Peer-Knoten des Anwenders im Lizenz-Vertrauensagenten sendet eine Antwort an den Lizenzmanager mit dem Lizenzobjekt und den verschlüsselten Entschlüsselungsinformationen zurück. Dann sendet der Lizenzmanager eine Antwort an die Backend-Applikation mit dem Lizenzobjekt und den verschlüsselten Entschlüsselungsinformationen. Die Backend-Applikation durchläuft eine Schleife über alle ausgewählten Laserbearbeitungsmaschinen und sendet die Entschlüsselungsinformationen an das Kryptosystem der jeweiligen Maschine. Die Interaktion zwischen der Backend-Applikation und dem Kryptosystem einer Lasermaschine zur

Entschlüsselung und Nutzung der Technologiedaten wird im nächsten Abschnitt durch ein separates Diagramm (siehe Unterkapitel 5.1.3) erläutert. Ist die Lizenz ungültig oder liegt ein Verstoß gegen die Nutzungsbedingungen vor, gibt der Smart Contract des Kanals eine Fehlermeldung über eine ungültige Lizenz an den Peer Node des Anwenders zurück, der wiederum eine Antwort mit der Fehlermeldung an den Lizenzmanager sendet. Der Lizenz-Asset-Manager sendet eine Antwort mit der Fehlermeldung an die Backend-Applikation. Auf der Seite des Lizenzzugriffs wird auf dem autorisierten Konto eine Fehlermeldung angezeigt. Insgesamt stellt der Ablauf der Interaktionen zwischen dem autorisierten Konto und dem Smart Contract des Kanals sicher, dass nur autorisierte Anwender auf die erworbenen Technologiedaten zugreifen können und die Vertraulichkeit der Lizenzinformationen gewährleistet ist.

Sequenzdiagramm zur Funktion Get all licenses

Der Ablauf des Abrufs aller erworbenen Lizenzen ist in Abbildung 5-9 als ein Sequenzdiagramm dargestellt. Ein autorisiertes Konto ruft die Seite mit der Lizenzliste auf.

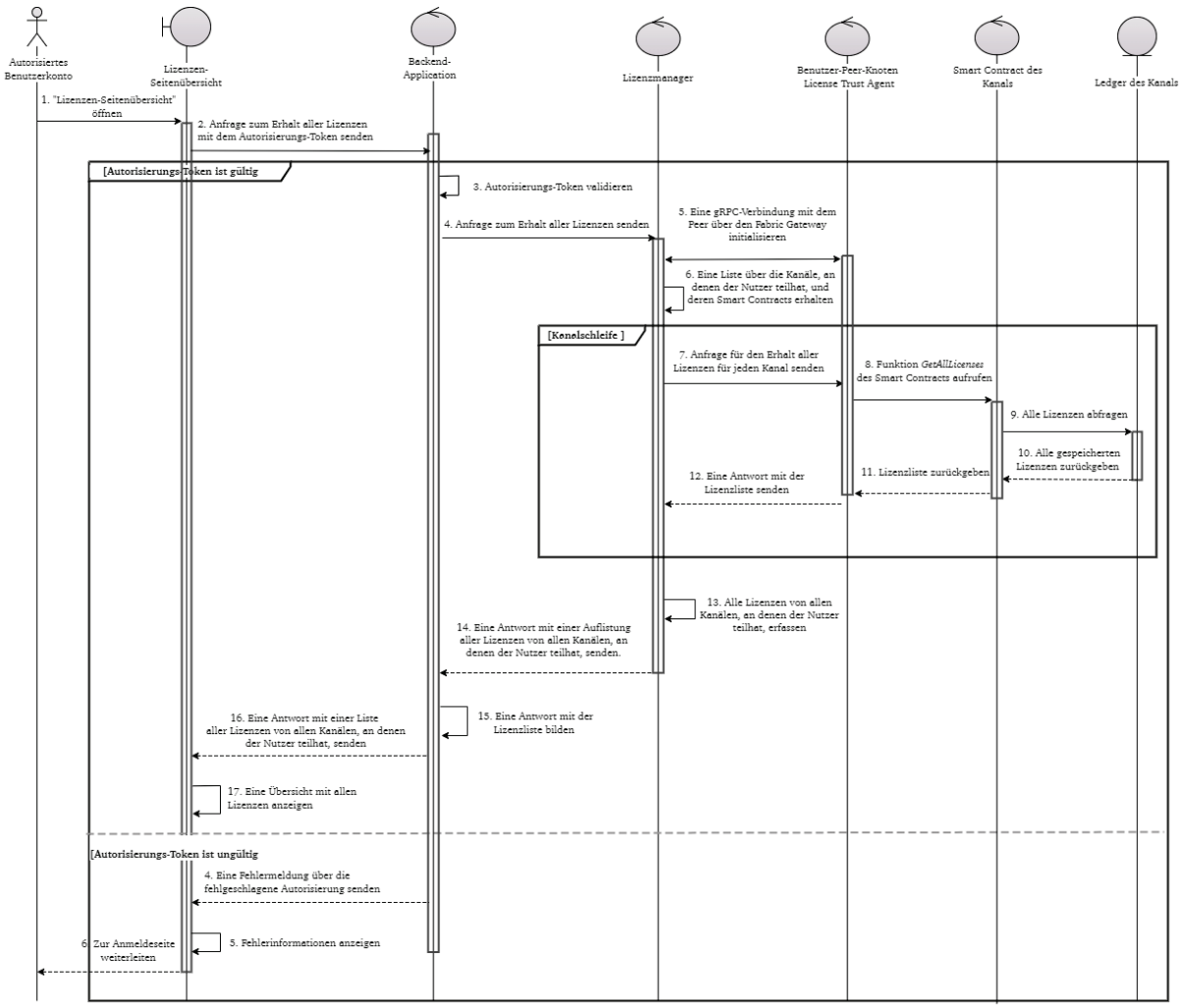

Abbildung 5-9: Sequenzdiagramm für Auflistung aller erworbenen Lizenzen (eigene Darstellung)

Die Seite sendet eine Anfrage an die Backend-Applikation, um alle erworbenen Lizenzen einzuholen. Die Backend-Applikation validiert das geschickte Autorisierungs-Token des autorisierten Kontos. Ist das Autorisierungs-Token gültig, sendet die Backend-Applikation eine Anfrage an den Lizenzmanager, um alle Lizenzen abzurufen. Der Lizenzmanager initialisiert dann eine *gRPC*-Verbindung mit den Peer-Knoten des Anwenders unter Verwendung des Anwenderzertifikats.

Der Lizenzmanager geht die Liste der verbundenen Kanäle durch und sendet eine Anfrage an den Peer-Knoten des Benutzers, um alle Lizenzen für jeden Kanal zu erhalten. Der Peer-Knoten des Anwenders ruft die Funktion *GetAllLicenses* im bereitgestellten Smart Contract auf, die alle Lizenzen aus dem Ledger der Kanäle abfragt. Der Lizenzmanager erfasst alle Lizenzen von allen Kanälen, erstellt eine Antwort und sendet sie an die Backend-Applikation. Die Backend-

Applikation sendet die Antwort an die Seite mit der Lizenzliste. Dort wird die Liste der Lizenzen für das autorisierte Konto angezeigt.

Ist das Autorisierungs-Token ungültig, sendet die Backend-Applikation eine Fehlermeldung an die Lizenzlistenseite, dass keine Autorisierung erfolgt ist. Die Seite Lizenzliste zeigt dem autorisierten Konto eine Fehlermeldung an und leitet den Nutzer zur Anmeldeseite des Systems weiter.

5.1.3 Implementierung des Kryptosystems

Das Kryptosystem ist eine Software, das auf dem Server, auf spezieller Hardware oder direkt auf der Laserbearbeitungsmaschine installiert werden kann. Diese kann alleinstehend laufen oder in das Softwaresystem der Laserbearbeitungsmaschine integriert werden. Das Kryptosystem ist für die Entschlüsselung der verschlüsselten Technologiedaten zuständig. Auf Basis der Technologiedaten-ID in den Lizenzinformationen ruft es die entsprechenden Technologiedaten ab, die bereits auf einem gemeinsamen Ordner innerhalb des Netzwerks abgelegt sind. Hierfür werden Entschlüsselungsinformationen benötigt – die verschlüsselten symmetrischen Schlüssel und der Verschlüsselungsalgorithmus.

In der prototypischen Implementierung fungiert ein *Raspberry Pi 4 Model B* als Maschinencomputer und Kryptosystem. Das verwendete Betriebssystem ist das *Raspberry Pi OS 64-bit* in der Version 6.1 [184]. Zusätzlich ist ein HTTP-Server darauf implementiert, um die Kommunikation mit anderen Systemkomponenten zu ermöglichen. Für die sichere Schlüsselverwaltung ist ein TPM physisch an die *GPIO-Pins* des *Raspberry Pi* angeschlossen. Es wird ein *OPTIGA™ TPM SLM 9670* verwendet [185]. Dieses TPM unterstützt die asymmetrische Kryptographie, einen RSA-Algorithmus mit Schlüssellängen bis zu 2048 Bit. Das TPM dient zur Entschlüsselung und Bereitstellung von symmetrischen Schlüsseln. Der Anwender besitzt bereits ein öffentliches/privates Schlüsselpaar. Auf dem TPM ist der private Schlüssel des Anwenders gespeichert. Sein öffentlicher Schlüssel steht dem Technologiedatenmarktplatz zur Verfügung, damit wird der symmetrische Schlüssel für die Verschlüsselung der Technologiedaten verschlüsselt. Mittels des *OpenSSL*-Werkzeugs wird ein RSA-Schlüsselpaar generiert, wobei der private Schlüssel des Anwenders auf dem TPM gespeichert wird. Die Abbildung 5-10 veranschaulicht die Schritte vom Zeitpunkt des Eingangs der Backendanfrage bis zur Bereitstellung der Technologiedaten für die Laserbearbeitungsmaschine.

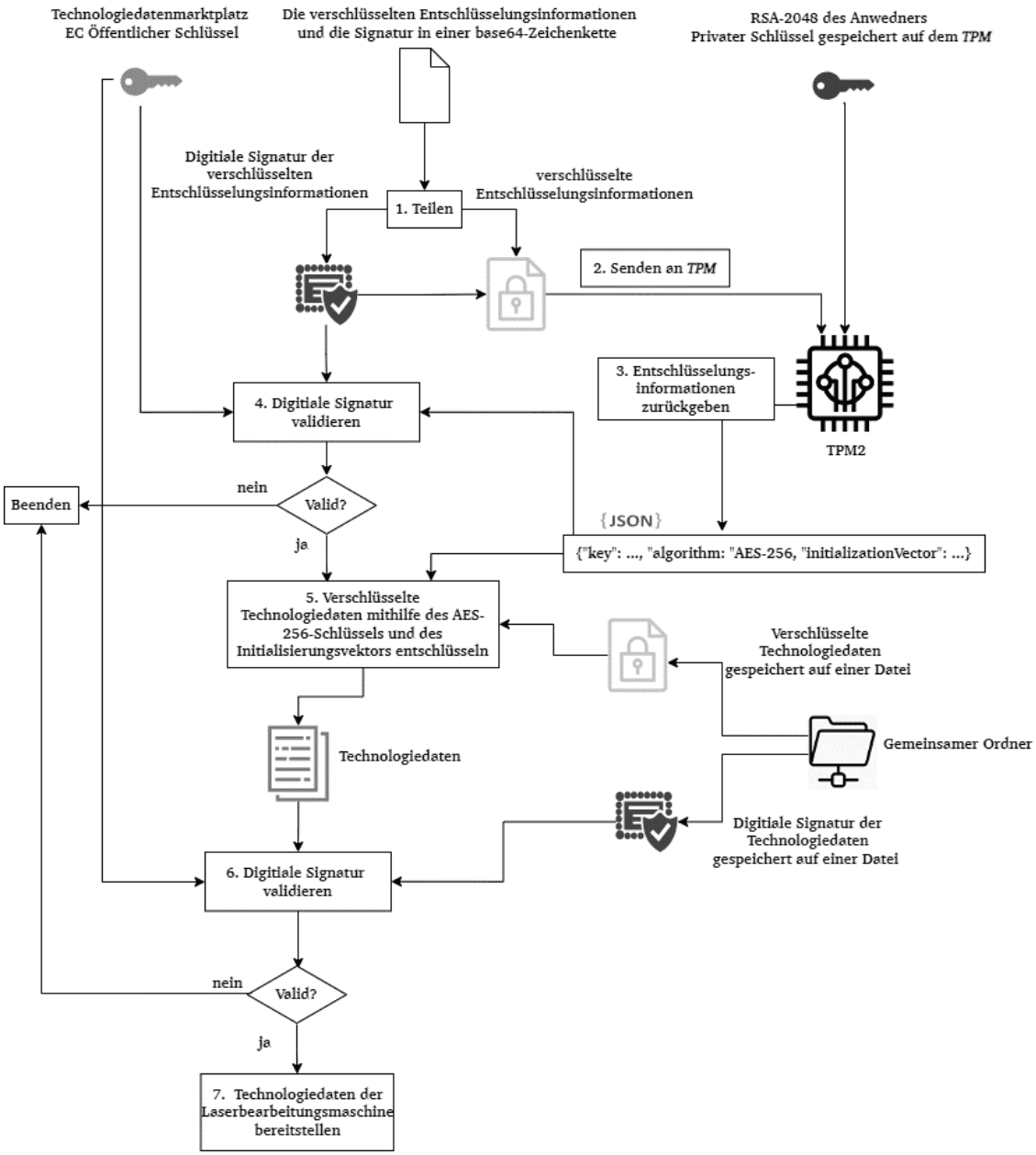

Abbildung 5-10: Kryptografische Operationen zur Entschlüsselung und Verifizierung des digitalen Signaturen (eigene Darstellung)

Die verschlüsselten Technologiedaten werden nach dem erfolgreichen Erwerb verschlüsselt, signiert und an den Anwender geschickt und schließlich auf einem gemeinsamen Ordner gespeichert. Das Kryptosystem ruft diese Daten anhand ihres eindeutigen Identifikators aus den Lizenzinformationen auf. Das Kryptosystem verfügt auch über eine Schnittstelle (API), über die sie Lizenzinformationen und verschlüsselte Entschlüsselungsinformationen von der Backend-Applikation erhält. Mit dem im TPM gespeicherten privaten Schlüssel entschlüsselt das Gerät die verschlüsselten Entschlüsselungsinformationen und gibt sie als Klartext an das Kryptosystem zurück. Das Kryptosystem nutzt dann den erforderlichen symmetrischen Schlüssel und den

Algorithmus zur Entschlüsselung der Technologiedaten. Schließlich werden die Technologiedaten der Laserbearbeitungsmaschine bereitgestellt.

Um den Entschlüsselungsvorgang der Technologiedaten zu starten, werden die verschlüsselten Entschlüsselungsinformationen zusammen mit den Lizenzinformationen vom Backend an das Kryptosystem gesendet. Die *do_POST*-Methode der Server-Klasse empfängt die Anfrage, die Lizenz wird dekodiert und auf das Vorhandensein der Felder *Entschlüsselungsinformationen* und *Technologiedaten-ID* geprüft. Sind beide Felder vorhanden, wird ein *TechnologyDataProvider*-Objekt mit der *Technologiedaten-ID* und den verschlüsselten Entschlüsselungsinformationen erstellt, und es wird ein *TPM_Gateway-Objekt* zur Entschlüsselung der verschlüsselten Entschlüsselungsinformationen mithilfe eines TPM-Geräts erzeugt.

Die entschlüsselten Entschlüsselungsinformationen werden anhand des öffentlichen Schlüssels des Technologiedatenmarktplatzes auf eine zulässige digitale Signatur geprüft. Wenn die Signatur gültig ist, wird das Entschlüsselungsinformationsobjekt aus der entschlüsselten Zeichenfolge geladen, und der symmetrische Schlüssel und der Initialisierungsvektor werden daraus extrahiert.

Dann werden die verschlüsselten Technologiedaten mit dem extrahierten symmetrischen Schlüssel und dem Initialisierungsvektor entschlüsselt. Außerdem wird die Herkunft der entschlüsselten Technologiedaten mit dem öffentlichen Schlüssel des Technologiedatenmarktplatzes verifiziert. Ist die Signatur gültig, wird eine Erfolgsrückmeldung an das Backend gesendet, die bestätigt, dass die Technologiedaten erfolgreich entschlüsselt wurden und dass die Laserbearbeitungsmaschine nun darauf zugreifen kann. Treten während des Entschlüsselungsvorgangs Fehler auf, wird eine interne Serverfehlermeldung an das Backend zurückgeschickt, zusammen mit einer Erklärung der Fehlerursache. Das folgende Codesegment illustriert die Implementierung der Funktion *do_POST* der Klasse *Server*:

```
def do_POST(self):
        content_length = int(self.headers['Content-Length'])
        data = self.rfile.read(content_length)
        try:
            _license = json.loads(data.decode('utf-8'))
            if 'decryptionInformation' in _license and 'technology_data_id' in _license:
                data_provider = TechnologyDataProvider(_license['technology_data_id'],
_license['decryptionInformation'])
                # device
                if osp.exists(TPM_DEVICE):
                    tmp_gateway = TPM_Gateway()
                # simulator
                else:
                    tmp_gateway = TPM_Gateway(port=2321)
                # decrypt the decryption information using TPM
```

```
                decryption_information =
tmp_gateway.decrypt(data_provider.encrypted_decryption_information)
                if not
data_provider.check_decryption_information_signature(str.encode(decryption_information)):
                    raise RuntimeError('Decryption information signature verification failed,
please contact the manufacturer.')
                # load the decryption information object
                decryption_information_obj = json.loads(decryption_information)
                if 'key' in decryption_information_obj and 'initializationVector' in
decryption_information_obj:
                    # decrypt the technology data file
                    symmetric_key =
base64.b64decode(str.encode(decryption_information_obj['key']))
                    iv =
base64.b64decode(str.encode(decryption_information_obj['initializationVector']))
                    decrypted_technology_data = data_provider.decrypt_data(symmetric_key, iv)
                # Verify the technology data signature
                if not data_provider.check_data_signature(decrypted_technology_data):
                    raise RuntimeError('Technology data signature verification failed, please
contact the manufacturer.')
                self.send_response(200)
                self.send_header('Content-type', 'text/plain')
                self.end_headers()
                self.wfile.write(b'Technology data file was decrypted successfully and accessed
by the laser machine.')
            else:
                raise RuntimeError('Invalid license: decryption information and technology data
id are missing, please contact the manufacturer.')
        except Exception as err:
            self.send_response(500)
            self.send_header('Content-type', 'text/plain')
            self.end_headers()
            self.wfile.write(str.encode(str(err)))
```

Prozedur 5-3: Codesegment für die Implementierung der Funktion do_POST der Klasse Server

Der private Schlüssel des Anwenders wird auf dem TPM während des gesamten Vorgangs sicher aufbewahrt. Der Entschlüsselungsprozess wird in Echtzeit ausgeführt, ohne dass die entschlüsselten Technologiedaten in der Datenbank gespeichert werden. Das folgende Codesegment zeigt die Implementierung der Entschlüsselungsfunktion im *TPM_Gateway*. Das TPM entschlüsselt den *base64*-verschlüsselten String, der die Entschlüsselungsinformationen enthält, und gibt einen String zurück, der ein *JSON*-Objekt darstellt, in dem ein symmetrischer Schlüssel, ein Initialisierungsvektor und ein Algorithmus enthalten sind.

```
def decrypt(self, encrypted_decryption_information: bytes) -> str:
        try:
            # Create a TPMT_RSA_DECRYPT structure
            scheme = pytss.TPMT_RSA_DECRYPT(scheme=pytss.constants.TPM2_ALG.OAEP,
details=pytss.TPMU_ASYM_SCHEME(oaep=pytss.TPMS_SCHEME_HASH(hashAlg=pytss.constants.TPM2_ALG_ID.SH
A256)))
            # decrypt
            decrypted_object = self.ectx.rsa_decrypt(self.handle,
encrypted_decryption_information, scheme)
            # get string from the TPMB_PUBLIC_KEY_RSA structure
            decrypted_string = decrypted_object.marshal().decode('ascii')
            # get the decryption information string
            decryption_information = re.sub(r'^.*?{', '{', decrypted_string).strip('\n')
            return decryption_information
```

```
        except Exception as e:
            raise e
```

Prozedur 5-4: Codesegment für die Entschlüsselungsfunktion im TPM

Die Validierung der digitalen Signatur sowohl für die Entschlüsselungsinformationen als auch für die Technologiedaten unter Verwendung des öffentlichen Schlüssels des Technologiedatenmarktplatzes ist ein entscheidender Schritt, um die Authentizität der Daten zu prüfen. Die Methode *validate_signature* in der Klasse *TechnologyDataProvider* ist eine Hilfsmethode, die als Eingabe die Signatur und Daten annimmt und dann die Signatur verifiziert. Sie gibt einen booleschen Wert zurück, der aussagt, ob die Signatur gültig ist oder nicht. Die Methode *check_decryption_information_signature* validiert die Signatur der entschlüsselten Entschlüsselungsinformationen und die Methode *check_data_signature* validiert die Signatur der entschlüsselten Technologiedaten. Der folgende Codeausschnitt zeigt die Implementierung der Methode *validate_signature:*

```
def validate_signature(self, signature: bytes, data: bytes) -> bool:
        manufacturer_signing_public_key_file = osp.join(settings.KEYS_DIR,
settings.MANUFACTURER_PUBLIC_KEY)
        if not osp.exists(manufacturer_signing_public_key_file):
            raise ValueError('The manufacturer\'s public key file must exist.')

        # Load the manufacturer's EC public key
        with open(manufacturer_signing_public_key_file, 'rb') as f:
            manufacturer_public_key = serialization.load_pem_public_key(f.read())
        # Verify the decryption information signature
        try:
            manufacturer_public_key.verify(
                signature,
                data,
                ec.ECDSA(hashes.SHA256())
            )
            return True
        except InvalidSignature:
            return False
```

Prozedur 5-5: Codesegment für die Implementierung der Signaturverifizierung

Durch die oben implementierten Funktionen wird die Vertraulichkeit sowie die Integrität und Authentizität der Daten sichergestellt.

Sequenzdiagramm zur Entschlüsselung der Technologiedaten

Das Sequenzdiagramm, Abbildung 5-11, zeigt den Kommunikationsfluss zwischen den Komponenten des Systems, die an der Entschlüsselung der Technologiedaten für eine bestimmte Laserbearbeitungsmaschine beteiligt sind.

Der Prozess beginnt nach Erhalt der verschlüsselten Entschlüsselungsinformationen, wie im vorhergehenden Abschnitt erläutert wurde. Die Backend-Applikation läuft in einer Schleife über alle Lasermaschinen, die von dem autorisierten Konto ausgewählt wurden, und für jede Lasermaschine wird eine Anfrage mit der Lizenz und den verschlüsselten Entschlüsselungsinformationen an das Kryptosystem auf der Lasermaschine gesendet. Das Kryptosystem holt sich dann die verschlüsselten Technologiedaten aus dem freigegebenen gemeinsamen Dateisystem (engl. Shared File System). Danach sendet das Kryptosystem die verschlüsselten Entschlüsselungsinformationen an das TPM auf dem Lasergerät. Das TPM entschlüsselt dann die verschlüsselten Entschlüsselungsinformationen mit dem privaten Schlüssel des Anwenders und sendet die Entschlüsselungsinformationen im Klartext zurück an das Kryptosystem. Das Kryptosystem verwendet dann die entschlüsselten Informationen, um die verschlüsselte Technologiedaten-Datei zu entschlüsseln. Im Anschluss stellt sie diese Daten der Lasermaschine zur Verfügung. Zum Schluss sendet das Kryptosystem eine Erfolgsmeldung an die Backend-Applikation, wonach der Vorgang erfolgreich abgeschlossen wurde.

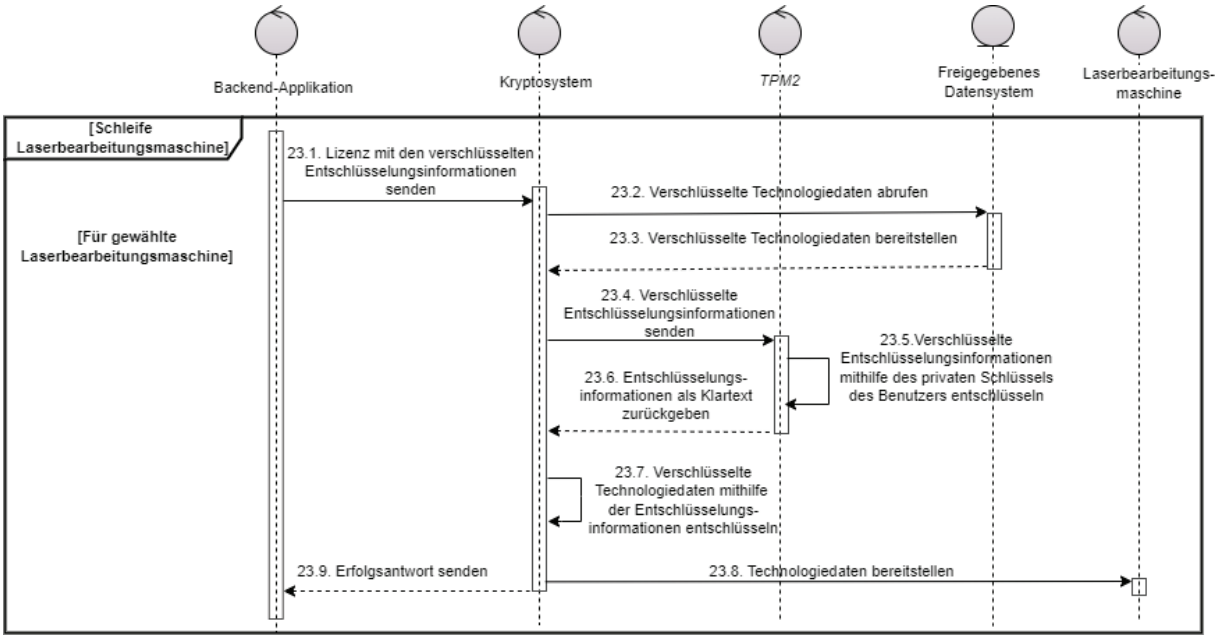

Abbildung 5-11: Sequenzdiagramm für Entschlüsselung von Technologiedaten (eigene Darstellung)

5.1.4 Implementierung des Assistenzsystems zur Lizenzmodellauswahl

Die *Python Knowledge Engine* (PyKE) ist eine wissensbasierte Inferenzmaschine (engl. Inference Engine) für *Python*, die für die Entwicklung regelbasierter Assistenzsysteme und wissensbasierter Anwendungen verwendet werden kann. Sie basiert auf dem RETE-Algorithmus,

einem effizienten Algorithmus zum Mustervergleich für regelbasierte Systeme. Die *PyKE* bietet eine Schnittstelle zur Definition von Regeln und Bedingungen und kann zur Durchführung von Vorwärts- und Rückwärtsverkettungen verwendet werden. Außerdem bietet sie eine Reihe von Werkzeugen zum Austesten (engl. Debugging) und Prüfen von Regelsätzen. *PyKE* ist quelloffen (engl. Open source) und kann in einer Vielzahl von Anwendungen eingesetzt werden, z. B. in Systemen zur Entscheidungsunterstützung, Diagnostik und Robotik [186].

Für die Implementierung des Assistenzsystems in dieser Dissertation wird *PyKE* verwendet, damit auf der Grundlage der eingegebenen Auftragsdaten die passenden Lizenzmodelle ausgewählt werden. Die erforderlichen Regeln und Fragen werden in den folgenden Dateien definiert:

- Die Datei *questions.kqb* definiert eine Reihe von Fragen, die dazu dienen, Eingaben des Anwenders zu erfassen. Die Fragen beziehen sich auf die festgelegten Auswahlkriterien für das geeignete Lizenzmodell, wie z. B. das Auftragsvolumen sowie die Flexibilität und Häufigkeit der Aufträge.

- Die Datei *rules_questions.krb* enthält die definierten Regeln und Bedingungen für die Auswahl der geeigneten Lizenzmodelle. Hierbei werden Regeln für die Auswahl der geeigneten Art, des Typs und der Metrik einer Lizenz auf der Basis der Anwendereingaben bestimmt, die als Antworten auf die festgelegten Fragen in der Datei *questions.kqb* gespeichert werden.

Auf diese Art und Weise kann das Assistenzsystem als eine Befehlszeilenanwendung eingesetzt werden, die Fragen stellt und die Antworten des Anwenders erhält, um das geeignete Lizenzmodell auszuwählen. Die Datei *driver.py* definiert die Funktionen zur Durchführung von Inferenzen auf der Grundlage der Eingaben des Anwenders. Die Funktion *doing_proof* nimmt die Regeln als Eingang (engl. Input) und führt eine Inferenz durch, um das geeignete Lizenzmodell auszuwählen und als Ergebnis (engl. Output) dem Anwender bereitzustellen. Die Funktion *questions_based_proving* führt Schlussfolgerungen auf der Grundlage der Anwendereingaben durch.

Es folgt ein Beispiel für die Verwendung des Assistenzsystems als Befehlszeilenanwendung. In diesem Beispiel werden auf Basis der eingegebenen Auftragsinformationen Antworten zu folgenden Fragen gegeben:

- Frage 1: Wie wird die Zeitkritikalität des Auftrags bewertet? Antwort: dringend.
- Frage 2: Wie umfangreich ist das Auftragsvolumen? Antwort: groß.
- Frage 3: Wie häufig wird der Auftrag erteilt? Antwort: oft.
- Frage 4: Wird der Auftrag regelmäßig erteilt? Antwort: ja.
- Frage 5: Gibt es Einschränkungen in Bezug auf: Mitarbeiter, Maschine, Fabrik oder Standort? Antwort: bestimmte Mitarbeiter.

Basierend auf diesen Eingaben wählt das System den passenden Wert für jedes Attribut des Lizenzmodells aus (Lizenzart: Mehrplatz, Lizenztyp: nutzergebunden, Lizenzmetrik: nutzungsunbegrenzt).

```
doing proof
_____________________________________________________________________________
When should the order be delivered?
  1. Urgent
  2. Not urgent
? [1-2] 1
model art: multiple_places
_____________________________________________________________________________
How flexible is the order? is there any restrictions regarding: User, machine, factory or
location?
  1. Specific Employee
  2. Specific Laser Machine
  3. Specific Plant
  4. Specific Location
? [1-4] 1
model type: user_based
_____________________________________________________________________________
How is the order volume?
  1. Small
  2. Large
? [1-2] 2
_____________________________________________________________________________
How often does the order come?
  1. Rarely
  2. Often
? [1-2] 2
_____________________________________________________________________________
How frequent does the order come?
  1. Frequent
  2. Not Frequent
? [1-2] 1
model metric: usage_unlimited

done
```

Prozedur 5-6: Beispiel für die Verwendung des Assistenzsystems als Befehlszeilenanwendung

Das bereits erwähnte Beispiel stellt die Implementierung des Assistenzsystems als eigenständige Anwendung dar. Für die Integration des Assistenzsystems in das Gesamtsystem muss eine einfachere Methode zur Eingabe der benötigten Informationen entwickelt werden die es dem Anwender erlaubt, die festgelegten Fragen über eine Webschnittstelle zu beantworten. Mit *PyKE* ist es möglich, eine Reihe von Fakten zu definieren, aus denen sich das Ergebnis für die Lizenzmodellauswahl ableiten lässt. Zu diesem Zweck werden die Regeln entsprechend geändert, so dass Fakten akzeptiert, statt dass Fragen gestellt werden. Die Anpassungen umfassen die folgenden Punkte:

- Die Datei *rules.krb* wird hinzugefügt, in der die Regeln und Bedingungen für die Auswahl des geeigneten Lizenzmodells definiert werden. Sie definiert Regeln für die Auswahl der geeigneten Art, des Typs und der Metrik auf der Grundlage der angegebenen Fakten. Dabei werden auch Bedingungen zur Überprüfung definiert, ob zum Beispiel der Auftrag dringend ist, ob das Auftragsvolumen klein oder groß ist und ob der Auftrag selten oder häufig vorkommt.

- Die Datei *models.py* wird hinzugefügt, in der die Klasse *LicenseModel* definiert ist, welche das ausgewählte Lizenzmodell darstellt. Die Klasse hat Attribute für die Art, den Typ und die Metrik des Lizenzmodells. Die Datei definiert auch Objekte für die verschiedenen Optionen im Zusammenhang mit den Fakten, die zur Auswahl des entsprechenden Lizenzmodells verwendet werden. In dieser Dabei sind auch die Klassen *Fact* und *FactRepository* definiert, die zur Darstellung der im Inferenzprozess verwendeten Fakten dienen.

- Die Funktion *facts_based_proving* wurde der *Driver*-Datei hinzugefügt. Sie nimmt ein *FactRepository*-Objekt als Eingabe und führt auf der Grundlage der bereitgestellten Fakten eine Inferenz durch.

Im Folgenden wird die Datei *questions.kqb* in der Prozedur 5-7 dargestellt. Die Datei *rules_questions.krb* wird in der Prozedur 5-8 dargestellt. Schließlich wird die Datei *rules.krb* in der Prozedur 5-9 dargestellt.

```
# questions.kqb

is_urgent($ans)
          When should the order be delivered?
          ---
          $ans = select_1
                    1: Urgent
                    2: Not urgent
```

```
which_type($ans)
          How flexible is the order? is there any restrictions regarding: User, machine, factory
or location?
          ---
          $ans = select_1
                    1: Specific Employee
                    2: Specific Laser Machine
                    3: Specific Plant
                    4: Specific Location

is_large($ans)
          How is the order volume?
          ---
          $ans = select_1
                    1: Small
                    2: Large

is_often($ans)
          How often does the order come?
          ---
          $ans = select_1
                    1: Rarely
                    2: Often

is_frequent($ans)
          How frequent does the order come?
          ---
          $ans = select_1
                    1: Frequent
                    2: Not Frequent
```

Prozedur 5-7: Darstellung der questions.kqb-Datei

```
# rules_questions.krb

art_multiple_places
    use which_art(multiple_places)
    when
        questions.is_urgent($ans)
        check $ans in (1,)

art_single_place
    use which_art(single_place)
    when
        questions.is_urgent($ans)
        check $ans in (2,)

type_user_based
    use which_type(user_based)
    when
        questions.which_type($ans)
        check $ans in (1,)

type_hardware_based
    use which_type(hardware_based)
    when
        questions.which_type($ans)
        check $ans in (2,)

type_floating
    use which_type(floating)
    when
        questions.which_type($ans)
        check $ans in (3,)

type_location_based
```

```
    use which_type(location_based)
    when
        questions.which_type($ans)
        check $ans in (4,)

metric_usage_limited
    use which_metric(usage_limited)
    when
        questions.is_large($is_large)
        questions.is_often($is_often)
        check $is_large in (1,) or $is_often in (1,)

metric_usage_unlimited
    use which_metric(usage_unlimited)
    when
        questions.is_large($is_large)
        check $is_large in (2,)
        questions.is_often($is_often)
        check $is_often in (2,)
        questions.is_frequent($is_frequent)
        check $is_frequent in (1,)

metric_time_limited
    use which_metric(time_limited)
    when
        questions.is_large($is_large)
        check $is_large in (2,)
        questions.is_often($is_often)
        check $is_often in (2,)

        questions.is_frequent($is_frequent)
        check $is_frequent in (2,)
```

Prozedur 5-8: Darstellung der rules_questions.krb-Datei

```
# rules.krb

art_multiple_places
    use which_art(multiple_places)
    when
        facts.is_urgent($is_urgent)
        check $is_urgent == 'Urgent'

art_single_place
    use which_art(single_place)
    when
        facts.is_urgent($is_urgent)
        check $is_urgent == 'Not urgent'

type_user_based
    use which_type(user_based)
    when
        facts.which_type($type)
        check $type == 'Specific employee'

type_hardware_based
    use which_type(hardware_based)
    when
        facts.which_type($type)
        check $type == 'Specific laser machine'

type_floating
    use which_type(floating)
    when
        facts.which_type($type)
        check $type == 'Specific plant'
```

```
type_location_based
    use which_type(location_based)
    when
        facts.which_type($type)
        check $type == 'Specific location'

metric_usage_limited
    use which_metric(usage_limited)
    when
        facts.is_large($is_large)
        facts.is_often($is_often)
        check $is_large == 'Small' or $is_often == 'Rarely'

metric_usage_unlimited
    use which_metric(usage_unlimited)
    when
        facts.is_large($is_large)
        check $is_large == 'Large'
        facts.is_often($is_often)
        check $is_often == 'Often'
        facts.is_frequent($is_frequent)
        check $is_frequent == 'Frequent'

metric_time_limited
    use which_metric(time_limited)
    when
        facts.is_large($is_large)
        check $is_large == 'Large'
        facts.is_often($is_often)
        check $is_often == 'Often'
        facts.is_frequent($is_frequent)
        check $is_frequent == 'Not frequent'
```

Prozedur 5-9: Darstellung der rules.krb-Datei

Zur Einbindung des Assistenzsystems in die Backend-Applikation wurde eine Webschnittstelle entwickelt, um die Antworten der Anwender zu sammeln und an die Backend-Applikation zu übermitteln. Anschließend verwendet die Backend-Applikation die Klassen *Fact* und *FactRepository*, um die Antworten in Fakten umzuwandeln, und ruft die Funktion *facts_based_proving* auf, um das geeignete Lizenzmodell auf der Grundlage der Anwendereingaben zu ermitteln.

5.1.5 Implementierung des Backends

Für die Implementierung des Backends stehen verschiedene Programmiersprachen und Werkzeuge zur Auswahl. *Python* stellt jedoch wegen ihrer Einfachheit, Vielseitigkeit und Verbreitung zwischen Entwicklern eine gute und geeignete Option für die Implementierung in dieser Dissertation dar. *Python* verfügt über eine Reihe von beliebten Frameworks für die Entwicklung von Webapplikationen, darunter *Flask* [187], *Pyramid* [188] und *Django* [189].

Flask ist ein einfaches und flexibles Framework, das sich für kleine bis mittelgroße Applikationen gut empfiehlt [190]. Es zeichnet sich durch seine Einfachheit und Benutzerfreundlichkeit aus. Darüber hinaus ermöglicht es Entwicklern, ihre eigenen Werkzeuge und Bibliotheken für verschiedene Zwecke einzusetzen. *Pyramid* ist ebenfalls ein flexibles und erweiterbares Framework, das eine Reihe von Funktionen für die Entwicklung großer und komplexer Anwendungen bereitstellt [190]. Außerdem bietet *Pyramid* zuverlässige und robuste Authentifizierungs- und Sicherheitsfunktionen. *Django* ist ein beliebtes und weit verbreitetes Framework, das eine umfassende Lösung für die Entwicklung von Webapplikationen bereitstellt. Es verfügt über integrierte Funktionen für die Authentifizierung, die Datenbankverwaltung und die URL-Weiterleitung und bietet eine leistungsstarke Oberfläche für die Verwaltung von Anwendungsdaten. *Django* ist außerdem bekannt für seinen hohen Sicherheitsgrad und die Fähigkeit, große Datenverkehrsvolumen zu bewältigen [190]. Aufgrund der umfassenden Funktionen und des starken Sicherheitsschwerpunkts ist *Django* eine beliebte Wahl für die Entwicklung komplexer Webapplikationen. Darüber hinaus ist *Django* dank seiner aktiven Entwicklergemeinschaft und umfangreichen Dokumentation leicht zu erlernen und zu verwenden.

Für das Backend wird in dieser Dissertation das *Django*-Framework in Verbindung mit weiteren Bibliotheken genutzt. Als Datenbankserver wird auf *PostgreSQL* zurückgegriffen [191]. Um eine *RESTful*-API bereitzustellen, die vom Frontend genutzt werden kann, verwendet das Backend das *Django-REST*-Framework [192].

Das Backend implementiert Funktionen zur Authentifizierung und zur Autorisierung von Benutzern, um das System gegen unautorisierten Zugriffe zu sichern. Im Folgenden wird das Sequenzdiagramm zum Anmeldeprozess vorgestellt, das diese Authentifizierungsfunktion darstellt.

Sequenzdiagramm zum Anmeldeprozess

Das in Abbildung 5-12 dargestellte Sequenzdiagramm illustriert den Anmeldeprozess eines registrierten Kontos über die Anmeldeseite der Webapplikation. Hierbei wird das Hauptziel verfolgt, die Identität des Nutzers durch die Autorisierung seiner eingegebenen Daten zu bestätigen. Der Zugang zur Applikation und ihren Funktionen, sowie den dazugehörigen Daten, ist ausschließlich autorisierten Nutzern vorbehalten.

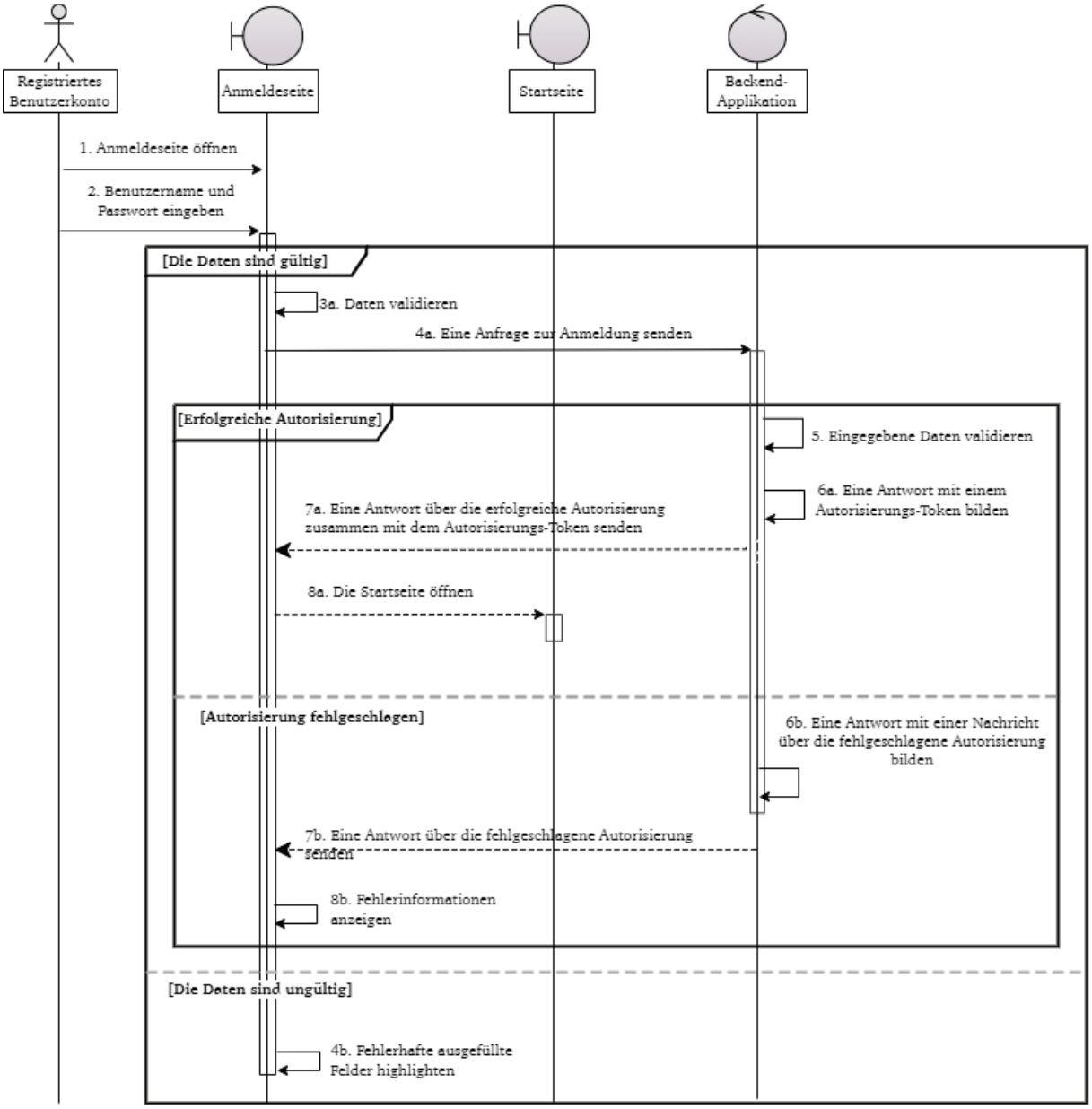

Abbildung 5-12: Sequenzdiagramm für den Anmeldeprozess (eigene Darstellung)

Der Anmeldevorgang wird eingeleitet, wenn ein Nutzer, der über ein registriertes Konto verfügt, seinen Benutzernamen und sein Passwort auf der Anmeldeseite eingibt. Nach der Validierung dieser Daten durch die Anmeldeseite wird eine Anmeldeanforderung an das Backend der Applikation gesendet. Das Backend überprüft diese Anforderung, und bei positivem Ergebnis erfolgt eine Rückmeldung über die erfolgreiche Autorisierung inklusive eines Autorisierungs-Tokens. Anschließend wird der Nutzer zur Startseite der Applikation weitergeleitet.

Falls der eingegebene Benutzername und/oder das Passwort nicht korrekt sind, gibt das Backend eine Rückmeldung über eine fehlgeschlagene Autorisierung aus und dem Nutzer wird auf der Anmeldeseite ein entsprechender Fehler angezeigt.

5.1.6 Implementierung des Frontends

Das Frontend ist der für den Nutzer sichtbare Bereich einer Webapplikation. Es handelt sich um die Benutzeroberfläche, die im Webbrowser angezeigt wird. Das Frontend schließt vom Layout und Design der Seite bis hin zu den Schaltflächen, Formularen und den anderen interaktiven Elementen alles ein. Letztlich handelt es sich also um ein grafische Aufarbeitung und Visualisierung der Daten. Hierfür soll eine Schnittstelle zur Datenkommunikation mit dem Backend zur Verfügung gestellt werden.

Diese grafische Benutzeroberfläche stellt den Anwendern die folgenden Interaktionen mit den Systemfunktionalitäten:

- Die Authentifizierung der Benutzer.
- Die Auswahl der geeigneten Laserbearbeitungsmaschinen zur Bearbeitung eines neuen Kundenauftrags.
- Die Lizenzmodellauswahl zur effektiven und flexiblen Laserbearbeitung.
- Die Bereitstellung von erworbenen Technologiedaten und Lizenzen.
- Die Übersicht über erworbene Lizenzen und deren Nutzungsstatus.
- Die Eintragung von neuen Laserbearbeitungsmaschinen im System.

Zur Durchführung eines neuen Laserbearbeitungsprozesses benutzt der Anwender die entwickelte Applikation, um ihn die ihn beim Erwerb der Technologiedaten sowie bei der Maschinen- und Lizenzmodellauswahl unterstützt. Er ruft zunächst die Startseite der Applikation auf, wobei er sich authentifizieren muss. Dafür gibt er seinen selbstgewählten Benutzernamen und sein Passwort ein und meldet sich an. Dieser Schritt ist für die Implementierung der in der Konzeption definierten Sicherheitsmaßnahmen von großer Bedeutung. Hier werden die beim Anmeldeprozess angegebenen Daten mit den hinterlegten

Daten verglichen, um die Identität des Nutzers nachzuweisen. Dadurch wird sichergestellt, dass nur berechtigte Personen Zugriff haben auf die Technologiedaten. Nach einer erfolgreichen Anmeldung wird die Startseite der Applikation angezeigt. Dort kann der Benutzer – wie in der Folge gezeigt wird – zwischen verschiedenen Optionen wählen.

Das Frontend der entwickelten Applikation ist entsprechend der in Abbildung 5-13 dargestellten Struktur aufgebaut.

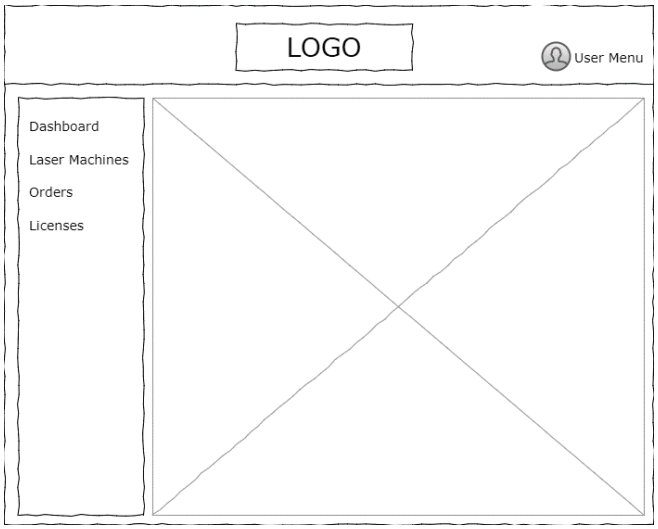

Abbildung 5-13: Struktur des Frontendes (eigene Darstellung)

Das Frontend hat drei Hauptkomponenten:

1. Die Obere Leiste: Die horizontale Leiste, die sich über den oberen Teil des Bildschirms erstreckt, enthält das Logo der Applikation sowie Informationen zum Benutzerkonto.

2. Die Seitenmenü: Das vertikale Menü befindet sich auf der linken Seite des Bildschirms. Es enthält Verknüpfungen zu verschiedenen Anwendungsbereichen und ermöglicht dem Benutzer, zu anderen Seiten zu navigieren, z. B. zur Seite mit den Lasergeräten oder den Lizenzen.

3. Das Hauptbereich: Hier werden die meisten Inhalte der Anwendung angezeigt. Der Hauptbereich kann alles enthalten, von Tabellen und Formularen bis hin zu Diagrammen und Grafiken.

Die Stärke dieses Entwurfs liegt in der Übersichtlichkeit und der einfachen Handhabung der Applikation, wobei zugleich alle für den Hauptinhaltsbereich vorgesehenen Informationen

angezeigt werden können. Klickt der Anwender auf eine Verknüpfung im Seitenmenü, wird der entsprechende Inhalt in den Hauptbereich geladen, während die obere Leiste jederzeit sichtbar bleibt und einen einfachen Zugang zu anderen Teilen der Applikation bietet.

Das Frontend wird in der Regel mithilfe unterschiedlicher Webtechnologien programmiert. Es kommuniziert mit dem Backend über dessen APIs, um Daten abzurufen und anzuzeigen. Zur Implementierung des Frontends kommen die folgenden Technologien und Bibliotheken zum Einsatz:

- *TypeScript: TypeScript* ist eine typbasierte Obermenge von *JavaScript*, die der Sprache optionale statische Typen, Klassen und Schnittstellen hinzufügt. *TypeScript* hilft, Fehler bei der Kompilierung abzufangen, erleichtert die Refaktorierung des Quellcodes und verbessert die Entwicklungserfahrung insgesamt [193].

- *React: React* ist eine *JavaScript*-Bibliothek für die Erstellung von Benutzeroberflächen. Mit ihr können Entwickler wiederverwendbare Benutzeroberflächenkomponenten erstellen und den Zustand ihrer Anwendung auf eine deklarative und effiziente Art verwalten [194].

- *React-Admin: React-Admin* ist ein Frontend-Framework für die Entwicklung von Applikationen. Es bietet eine Reihe von vorgefertigten UI- Komponenten für übliche Aufgaben wie zum Beispiel Datenvisualisierung, Datenbearbeitung und Authentifizierung, was die Entwicklung komplexer Applikationen mit geringem Aufwand ermöglicht [195].

- *MUI: MUI* ist eine populäre *React*-Komponentenbibliothek, die vorgefertigte UI- Komponenten auf Basis des Material-Design-Systems bereitstellt. Sie bietet eine übersichtliche und moderne Designsprache mit dem Schwerpunkt auf Benutzerfreundlichkeit und Zugänglichkeit [196].

Die Abbildung 5-14 veranschaulicht die Hauptmodule des Frontends. Jedes Modul stellt verschiedene *React*-Komponenten für eine Entität zur Verfügung, z. B. enthält das Modul *laser_machines* die *React*-Komponenten zum Auflisten, Anzeigen, Erstellen und Löschen von Lasermaschinen. Die Pfeile in der Abbildung veranschaulichen die Abhängigkeiten zwischen den Modulen. Es ist zu erkennen, dass alle Module abhängig sind vom *authProvider*, da dieser

die Authentifizierungs- und Autorisierungsfunktionen bereitstellt. Eines der wichtigsten Module ist der *dataProvider*, der für das Abrufen der Daten aus dem Backend verantwortlich ist.

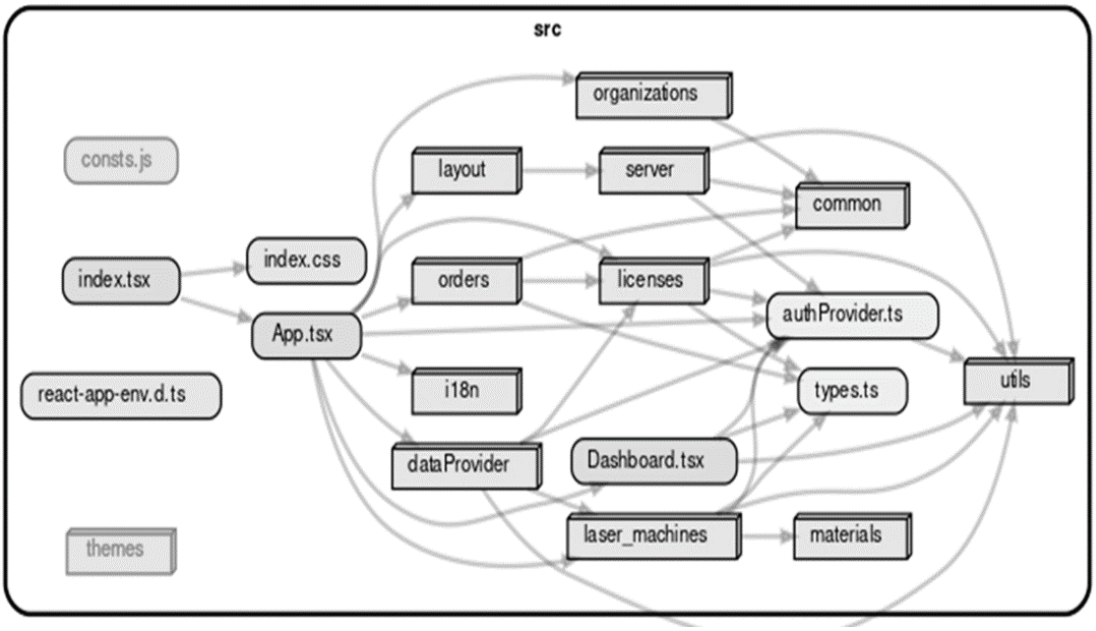

Abbildung 5-14: Hauptmodule des Frontends (eigene Darstellung)

Im Anhang 3 werden sämtliche Module, ihre *React*-Komponenten und deren Abhängigkeiten in einer detaillierten Abbildung dargestellt.

5.2 Fazit

Im Rahmen dieser Dissertation wurde ein Assistenzsystem zur Unterstützung von Anwendern bei der effektiven und sicheren Nutzung von Technologiedatenmarktplätzen in der Lasermarkierung prototypisch implementiert. Die Webapplikation bietet über ein systematisches Vorgehen die Möglichkeit, geeignete Laserbearbeitungsmaschinen für spezifische Laseraufträge auszuwählen. Ebenso dient das Assistenzsystem einer effizienten Auswahl von Lizenzmodellen für den Erwerb neuer Technologiedaten. Zusätzlich bietet die Webapplikation Funktionen zur sicheren Bereitstellung von Technologiedaten. Die grafische Benutzeroberfläche ermöglicht die Interaktion des Benutzers mit der Applikation. Darüber wird der Anwender aufgefordert, seine Anmeldedaten einzugeben, um seine Identität zu verifizieren.

Durch die Implementierung des Lizenz-Vertrauensagenten mithilfe der Blockchain- und Smart-Contract-Technologien wird eine sichere Nutzung von erworbenen Technologiedaten

gewährleistet. Hierbei wurde ein privates Blockchain-Netzwerk eingerichtet, in dem nur genehmigte Transaktionen von autorisierten Nutzern durchgeführt werden können. Für die Einrichtung des Blockchain-Netzwerks wurde das *Hyperledger Fabric*-Framework verwendet. Die intelligenten Verträge wurden auf Basis der im Konzeptkapitel definierten Lizenzmodelle umgesetzt. Bei der Implementierung der Verträge wurde auf die Programmiersprache *NodeJS* zurückgegriffen.

Durch die Implementierung des Lizenzmanagers könnten alle definierten Funktionen zur Interaktion einerseits mit dem Lizenz-Vertrauensagenten und andererseits mit dem Backend realisiert werden. Für die Implementierung wurden die Programmiersprachen *JavaScript* und *TypeScript* verwendet. In diesem Kapitel finden sich zwei Sequenzdiagramme, die den Ablauf des Zugriffs auf eine bestimmte Lizenz sowie auf alle erworbenen Lizenzen darlegen. Dank des implementierten Lizenzmanagers können die erworbenen Lizenzen erstellt, abgerufen und zugeordnet werden und ihre Nutzung ist rückverfolgbar. Das gewährleistet eine sichere Nutzung und eine lückenlose Nutzungskontrolle der erworbenen Technologiedaten, womit wesentliche Sicherheitsanforderungen erfüllt sind. Der Datensicherheit wurde sowohl durch Software- als auch Hardwarelösungen Genüge getan. Das Kryptosystem wurde auf einem *Raspberry Pi 4 Model B* implementiert, an das das Hardwaremodul *TPM SLM 9670* angeschlossen wurde. Das TPM bietet eine robuste und sichere Methode zur Verwaltung der privaten Schlüssel von Anwendern und stellt kryptographische Funktionen zur Datenentschlüsselung und Signaturüberprüfung bereit.

Die Bereitstellung der benötigten Technologiedaten zusammen mit passenden Lizenzmodellen basiert auf den Eingaben aus dem Auftrag und der Erfahrung des Anwenders. Dabei werden auf Grundlage des entwickelten Assistenzsystems systematisch einfache Fragen gestellt. Hierbei wurde die *Python Knowledge Engine* (PyKE) für die Implementierung der Inferenzmaschine verwendet. Die Wissensregeln für eine optimale Lizenzmodellauswahl wurden verfasst. Um das Assistenzsystem in das Gesamtsystem zu integrieren, wurde Anpassungen vorgenommen, indem die Regeln in Facts umgewandelt wurden. Nun können die Antworten vom Anwender verarbeitet und in ein passendes Format umgewandelt werden.

Die Datenverarbeitung sowie die Datenkommunikation werden über das Backend durchgeführt, das das Kernstück der Implementierung darstellt. Zur Erstellung des Backends wurde das *Django*-Framework zusammen mit weiteren Bibliotheken und *PostgreSQL* genutzt. Der Datenaustausch erfolgt mittels *JSON*-Dateien und über verschiedene APIs. Das Frontend wurde mit drei Hauptkomponenten aufgebaut: Obere Leiste, Hauptbereich und Seitenmenü. Für die

Implementierung kamen die Technologien und Frameworks: *TypeScript, React, React-Admin* und *MUI* zum Einsatz. Eine Übersicht über die im Rahmen der prototypischen Implementierung verwendeten Programmierwerkzeuge und Technologien bietet die Abbildung 5.15.

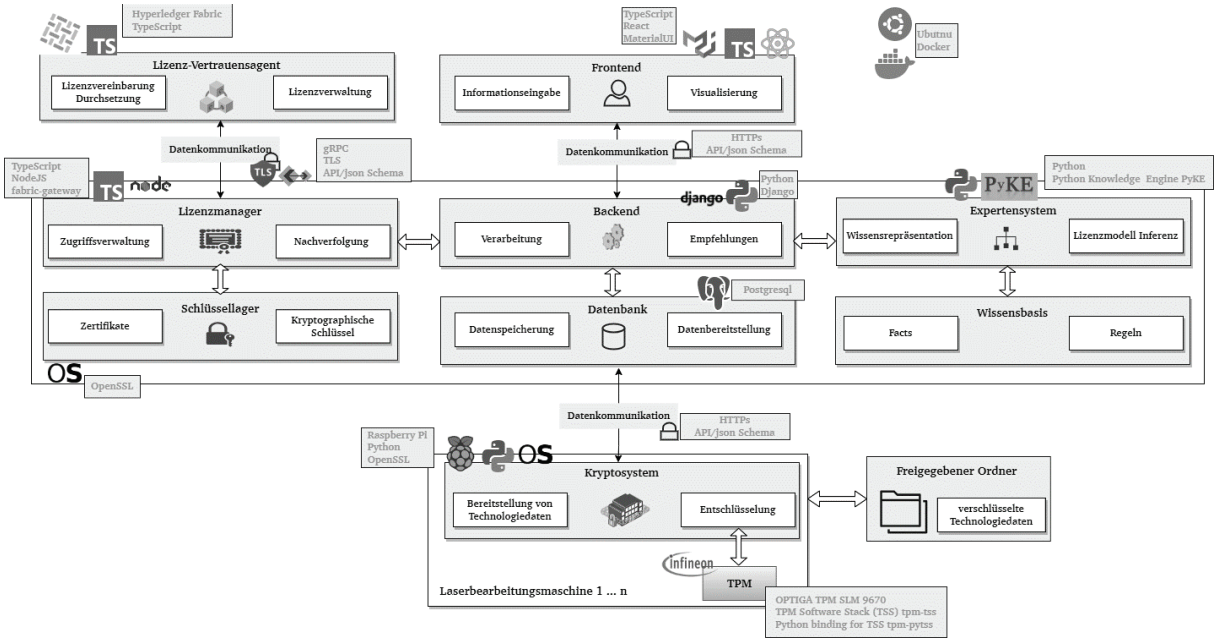

Abbildung 5-15: Verwendete Programmierwerkzeuge und Technologien (eigene Darstellung)

6 VALIDIERUNG UND VERIFIKATION

Aufbauend auf dem in Kapitel 4 entwickelten Konzept und der im letzten Kapitel dargelegten Implementierung erfolgt in diesem Kapitel die Validierung und Verifikation. Dabei werden die Tragfähigkeit des Konzeptes und die Funktionalität der prototypischen Implementierung geprüft. Hierfür wird ein repräsentativer Anwendungsfall in der Lasermarkierung ausgewählt. Die Verifikation wird im Anschluss an die Validierung durchgeführt. Dabei wird geprüft, ob das entwickelte System den zu Beginn formulierten Anforderungen genügt. Schließlich erfolgt ein Fazit zur Validierung und Verifikation.

6.1 Methodik zur Validierung und Verifikation

Der Begriff Validierung wird in der Norm DIN EN ISO 9000:2015-12 wie folgt definiert:

> ***Bestätigung durch Bereitstellung eines objektiven Nachweises, dass die Anforderungen für einen spezifischen beabsichtigten Gebrauch oder eine spezifische beabsichtigte Anwendung erfüllt worden sind [197].***

Bei der Validierung wird sichergestellt, dass ein System bestimmten Anforderungen und Kriterien entspricht. Beispielsweise kann durch Anwendertests geprüft werden, ob das entwickelte System seinen beabsichtigten Gebrauchszweck und die Zielvorgaben erfüllt. In der Norm DIN EN ISO 9000:2015-12 wird der Begriff Verifikation wie folgt definiert:

> ***Bestätigung durch Bereitstellung eines objektiven Nachweises, dass festgelegte Anforderungen erfüllt worden sind [197].***

Durch die Verifikation wird nachgewiesen, dass die festgelegten Anforderungen erfüllt worden sind [198]. Im Grunde genommen ist die Validierung die Überprüfung eines Systems in Bezug auf seine Tauglichkeit in Einsatzszenarien. Damit wird sichergestellt, dass das Endprodukt seinen vorgesehenen Zweck erfüllt und den Benutzeranforderungen entspricht. Wobei der Fokus bei der Verifizierung auf der Überprüfung eines Systems auf Erfüllung seiner festgelegten Anforderungen liegt.

Die Interaktion zwischen dem Anwender und der entwickelten und implementierten Webapplikation und Subsystemen wird in diesem Kapitel analysiert, um die Bedienbarkeit und Gebrauchstauglichkeit zu untersuchen.

Die Methode des *Cognitive Walkthrough* wird für die Validierung des entwickelten Konzeptes in der vorliegenden Dissertation verwendet. Bei dieser Methode wird der Prozess der Problemlösung von einem Nutzer durchgeführt. Hierdurch wird überprüft, ob der Nutzer die vom entwickelten System beabsichtigten Aufgaben ausführen kann. Dabei werden Nutzungsprobleme simuliert und durchgegangen, um Rückschlüsse zu ziehen bezüglich der Funktionen der Applikation [199].

Nun wird der repräsentative Anwendungsfall in der Lasermarkierung gewählt, wofür Technologiedaten auf dem Technologiedatenmarktplatz erworben werden sollen. Der Anwendungsfall wird Schritt für Schritt durchgeführt und untersucht, um dabei die Funktionen der Webapplikation dahingehend zu prüfen, ob sie ihre Aufgabe – die Unterstützung des Anwenders bei der effektiven und sicheren Nutzung von Technologiedatenmarktplätzen – erfüllen. Ziel ist es, dass die benötigten Technologiedaten zur Bearbeitung des Laserprozesses effektiv bereitgestellt werden. Außerdem soll die sichere Übermittlung und Nutzung von Technologiedaten gewährleistet werden.

6.2 Auswahl des repräsentativen Anwendungsfalls

Die Laserbearbeitung hat zu signifikanten Fortschritten im Bereich der Fertigung komplexer und präzisionsbedürftiger Bauteile erzielt. Diese Technologie ermöglicht nicht nur eine schnelle und präzise Verarbeitung, sondern eröffnet auch innovative Möglichkeiten zur Bearbeitung einer breiten Palette von Materialien mit hoher Qualität und Perfektion. Sie findet Anwendung in diversen Industriezweigen, einschließlich der Automobilindustrie und Medizintechnik, und ist sowohl für metallische als auch nichtmetallische Werkstoffe, wie beispielsweise Holz, geeignet.

Um in der Laserfertigung die spezifischen Anforderungen an Genauigkeit und Effizienz zu erfüllen, sind präzise Bearbeitungsbedingungen essentiell. Insbesondere die Parametrisierung des Lasersystems hat eine signifikante Auswirkung auf die Verarbeitungsqualität. Die hierfür erforderlichen spezialisierten Technologiedaten können Anwender auf speziellen Technologiedatenmarktplätzen erwerben. Die vorliegende Konzeption zielt darauf ab, den

Anwender in der Interaktion mit diesen Technologiedatenmarktplätzen zu unterstützen. Dies wird im Folgenden durch ein exemplarischer Anwendungsfall im Bereich des Lasermarkierverfahrens validiert.

Im Rahmen des entwickelten Systems werden spezifische Abläufe durch die drei zentralen Systemelemente zur Unterstützung des Anwenders bei der effektiven und sichern Nutzung von dem Technologiedatenmarktplatz ausgeführt, um die für die Ausführung von Fertigungsaufträgen benötigten Technologiedaten bereitzustellen. Im Kontext der Planungsphase werden initial Handlungsempfehlungen hinsichtlich der Auswahl von Laserbearbeitungsmaschinen und Lizenzmodellen bereitgestellt. In der Produktionsphase koordiniert das System die sichere Nutzung der Technologiedaten, die auf Basis der erworbenen Lizenzen gesteuert wird, und gewährleistet deren zeitgerechte Übermittlung an die zugehörigen Laserbearbeitungsmaschinen.

In der Validierung wird davon ausgegangen, dass der Anwender über eine Vielzahl von Laserbearbeitungsmaschinen verschiedener Hersteller verfügt, die für unterschiedliche Anwendungen, Technologietypen und Materialien eingesetzt werden. Um die Maschinen im System effektiv zu verwalten, ist ein initialer Registrierungsprozess der existierenden Laserbearbeitungsmaschinen erforderlich.

Über die Webapplikation kann der Anwender die Seite *Laser Machines* für die Verwaltung von Laserbearbeitungsmaschinen aufrufen. Der Anwender kann durch Betätigen des *Create*-Befehls eine neue Maschineninstanz im System anlegen. Hierfür müssen eine Reihe von Informationen erfasst werden, einschließlich Maschinenname, Hersteller, verwendete Lasertechnologie, Prozesstyp und maximale Laserleistung. Zusätzlich besteht die Möglichkeit, eine ausführliche Beschreibung und visuelle Darstellungen der Maschine einzufügen.

Ein signifikantes Detail in dieser Initialisierungsphase ist die Definition einer individuellen Identität für das Kryptosystem der jeweiligen Laserbearbeitungsmaschine, meist repräsentiert durch eine IP-Adresse. Diese dient als Empfängeradresse für die verschlüsselten Entschlüsselungsinformationen. Nach erfolgreicher Eintragung wird der Laserbearbeitungsmaschine eine spezifische Netzwerkadresse im Kryptosystem zugeordnet. Darüber hinaus bietet das System die Möglichkeit, eine Liste bearbeitbarer Materialien inklusive der maximalen Dicken zu speichern. Diese Informationen sind essentiell, um die richtigen Technologiedaten für diverse Laserbearbeitungsprozesse zu identifizieren und zur Verfügung zu stellen. Das dient der effizienten und schnellen Verwaltung der Laserbearbeitungsmaschinen.

Die Abbildung 6-1 illustriert den Prozess der Eingabe einer neuen Laserbearbeitungsmaschine, spezifisch des Modells *TruMark Station 3000* von der Firma *TRUMPF*. Die Informationen für die Maschineneintragung werden aus dem technischen Datenblatt des Herstellers entnommen [200]. In dieser Abbildung werden die notwendigen Attribute wie Maschinenname, Hersteller, Technologieart, Prozesstyp, maximale Laserleistung und weitere spezifische Merkmale dargestellt, die für die effiziente und präzise Durchführung der Laserbearbeitungsprozesse entscheidend sind.

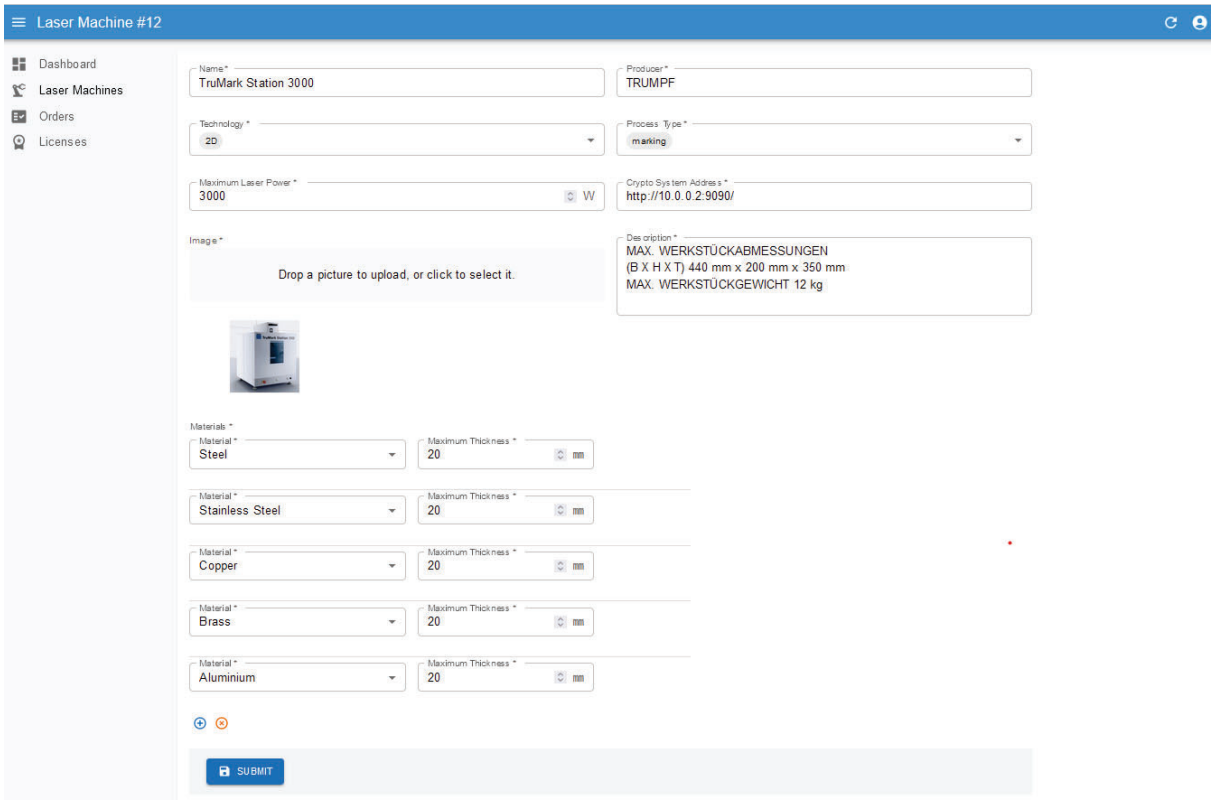

Abbildung 6-1: Eingabe einer neuen Laserbearbeitungsmaschine (eigene Darstellung)

In Abbildung 6-2 werden zusätzliche Laserbearbeitungsmaschinen diverser Hersteller und mit unterschiedlichen Prozess- und Technologietypen dargestellt, die im Besitz des Anwenders sind.

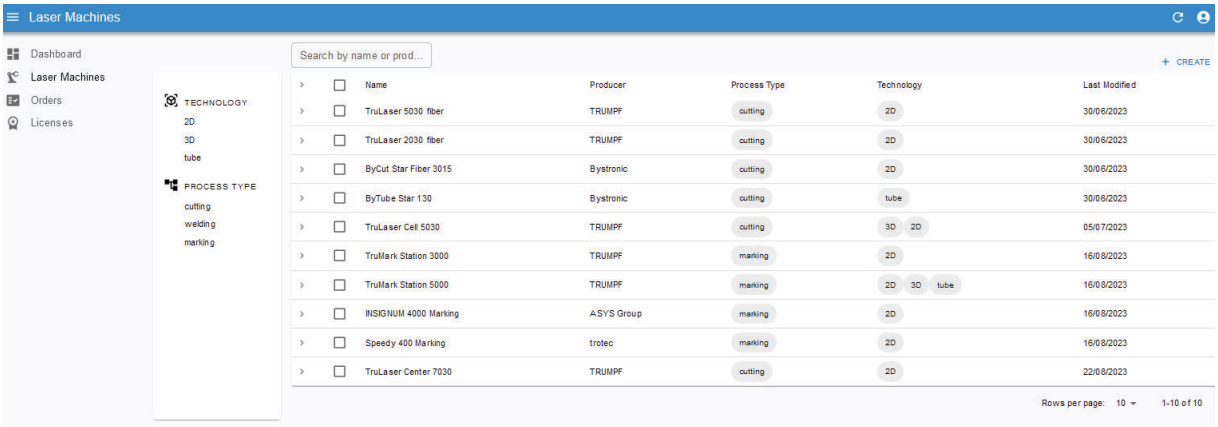

Abbildung 6-2: Alle verfügbaren Laserbearbeitungsmaschinen beim Anwender (eigene Darstellung)

Diese vielfältige Erfassung von Maschinen dient dazu, die Funktionalität der entwickelten Applikation umfassend zu evaluieren und die Robustheit und Flexibilität der Softwarelösung unter verschiedenen Betriebsbedingungen zu testen.

Vor der Durchführung der gewählten Anwendungsfälle ist es wichtig zu betonen, dass die Webanwendung, der Lizenzvertrauensagent und das Kryptosystem als integrale Bestandteile eines koordinierten Netzwerksystems fungieren. Die definierten Anwendungsfälle spiegeln den typischen intendierten Gebrauch des entwickelten Systems wieder.

Der Anwendungsfall, der zur Evaluierung der vielschichtigen Funktionalitäten des Systems dient, bezieht sich auf dem Lasermarkieren. Konkret wird eine Anfrage für die Bereitstellung von Technologiedaten für einen Kundenauftrag gestellt, der das Lasermarkieren von 2300 zweidimensionalen Edelstahlbauteilen umfasst. Zudem ist die Auslieferung der bearbeiteten Bauteile innerhalb einer Frist von einer Woche vorgesehen.

In der initialen Phase wählt der Anwender die Kategorie *Markieren* (engl. Marking) aus der Liste der verfügbaren Prozessarten aus. Anschließend präzisiert der Anwender unter der Rubrik *Technologie*, dass eine zweidimensionale Bearbeitung vorgesehen ist. Im nächsten Schritt spezifiziert er *Edelstahl* als zu bearbeitendes Material. Daraufhin erfolgt eine gezielte Auswahl von Lasermarkiermaschinen, die für die Markierung von zweidimensionalen Edelstahlbauteilen geeignet sind. Da die Materialdicke für den Markierungsprozess nicht relevant ist, wird das Eingabefeld für die Materialdicke in diesem speziellen Anwendungsfall nicht als obligatorisch gekennzeichnet. Gemäß den Empfehlungen des Assistenzsystems kommen vier Laserbearbeitungsmaschinen für den Markierungsvorgang von zweidimensionalen

Edelstahlbauteilen in Frage, von denen zwei identische Leistungsmerkmale aufweisen. Damit ist die Funktionalität des Systemelements für die Maschinenauswahl in diesem Anwendungsfall erfolgreich erfüllt. Die Abbildung 6-3 veranschaulicht das Ergebnis dieser Phase.

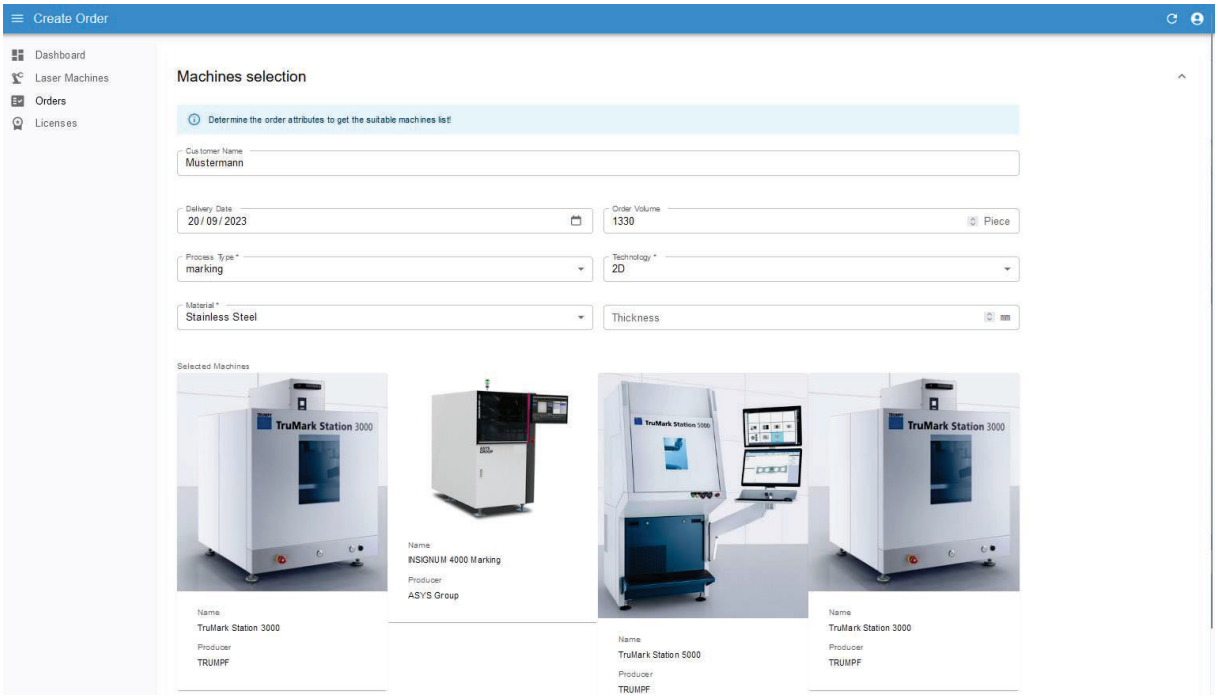

Abbildung 6-3: Ergebnis der Maschinenauswahl (eigene Darstellung)

In der nachfolgenden Phase werden dem Anwender Handlungsempfehlungen bezüglich geeigneter Lizenzmodelle für die Bereitstellung von Technologiedaten präsentiert. Um eine zielgerichtete Auswahl der Lizenzmodelle zu ermöglichen, wird dem Anwender eine Reihe von Fragen vorgelegt. Die Beantwortung dieser Fragen dient der Spezifikation der Attribute der gewünschten Lizenzmodelle. Im Folgenden werden die spezifischen Fragen sowie die Antworten des Anwenders dargelegt:

1. Erste Frage: Wie wird die Zeitkritikalität des Auftrags bewertet? Antwort: dringend.

2. Zweite Frage: Wie umfangreich ist das Auftragsvolumen? Antwort: groß.

3. Dritte Frage: Wie häufig wird der Auftrag erteilt? Antwort: nicht häufig.

4. Vierte Frage: Gibt es Einschränkungen in Bezug auf: Mitarbeiter, Laserbearbeitungsmaschine, Fabrik oder Standort? Antwort: keine Einschränkungen.

Basierend auf diesen Antworten empfiehlt das Assistenzsystem vier Lizenzmodelle, die unterschiedliche Kombinationen von Attributen aufweisen. Zu den empfohlenen Lizenzarten zählt *Mehrplatz*, während die Lizenztypen als *hardwaregebunden*, *nutzergebunden*, *floating* und *standortgebunden* klassifiziert werden. Als Lizenzmetrik wird *nutzungsbasiert* vorgeschlagen. Diese Empfehlungen werden in Abbildung 6-4 detailliert wiedergegeben.

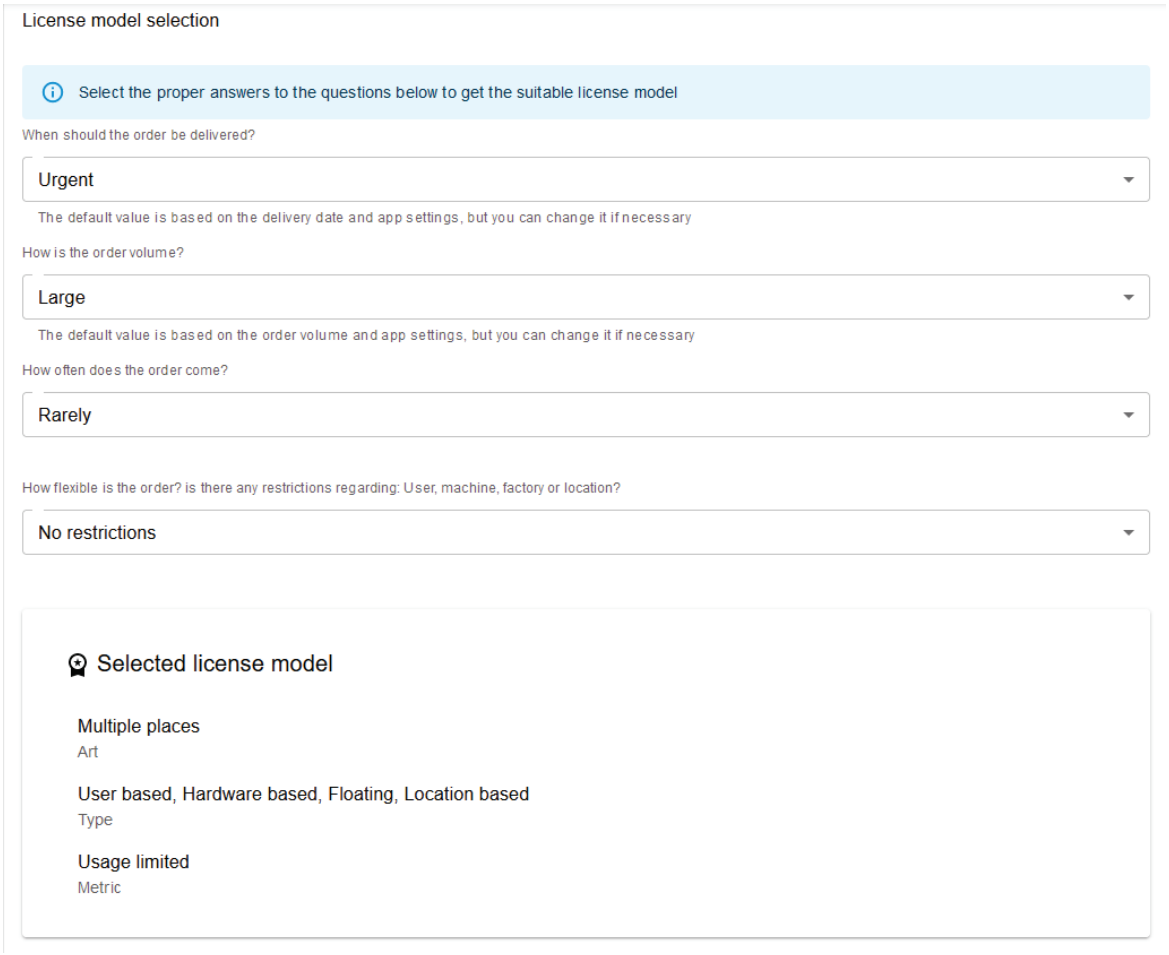

Abbildung 6-4: Antworten des Anwenders und empfohlene Lizenzmodelle (eigene Darstellung)

Die generierten Empfehlungen bezüglich der Auswahl von Laserbearbeitungsmaschinen und Lizenzmodellen können nun als CSV-Datei exportiert werden. Diese exportierte Tabelle dient der Auftragsplanung als Ressource für die Bereitstellung der erforderlichen Technologiedaten.

Im nächsten Schritt ist die Auswahl des geeigneten Lizenzmodells erforderlich. In der Auftragsplanung liegen sämtliche erforderliche Informationen vor, um eine fundierte Entscheidung bezüglich der Maschinenauswahl für den Auftrag zu treffen. Gemäß den

Handlungsempfehlungen des Assistenzsystems werden vier Lizenzmodelle als Optionen vorgeschlagen:

1. *Mehrplatz, hardwaregebunden, nutzungsbasiert*

2. *Mehrplatz, nutzergebunden, nutzungsbasiert*

3. *Mehrplatz, floating, nutzungsbasiert*

4. *Mehrplatz, standortgebunden, nutzungsbasiert*

Im Kontext des spezifischen Anwendungsfalls wird unter der Prämisse aktueller Gegebenheiten und vorhandener Informationen die zweite Option mit den Attributen *Mehrplatz, nutzergebunden, nutzungsbasiert* von der Auftragsplanung favorisiert. Außerdem wurde es beschlossen, den Auftrag auf den zwei identischen Laserbearbeitungsmaschinen des Modells *TruMark Station 3000* von *TRUMPF* zu bearbeiten.

Die getroffene Entscheidung wird zur weiteren Verarbeitung an die ERP-Abteilung übermittelt, um die notwendigen Technologiedaten vom entsprechenden Technologiedatenmarktplatz erwerben zu können. Eine Lizenz mit den festgelegten Attributen wird generiert und unter der ID *X53AfGII7yGARxqYXk9Y* im Lizenzvertrauensagenten für den Anwender hinterlegt. Da die Lizenz nutzergebunden ist, ist die Spezifizierung der E-Mail-Adressen der autorisierten Benutzer im Lizenzvertrag notwendig. Hierfür wird die E-Mail-Adresse beispielhaft für den Mitarbeiter *Martin* (martin@example.com) hinterlegt. Die betreffenden Technologiedaten werden in verschlüsselter Form an den Anwender versandt und anschließend in einem dafür freigegebenen Ordner abgelegt.

In der Produktionsphase findet erneut eine Interaktion mit der Webapplikation statt. In dieser Phase werden die Technologiedaten für die ausgewählten Laserbearbeitungsmaschinen bereitgestellt. Der Benutzer, in diesem Fall *Martin*, ist beauftragt, den Auftrag zu bearbeiten. Er loggt sich in die Webapplikation ein. Die Authentifizierung erfolgt durch die Eingabe seiner E-Mail-Adresse und seines Passworts (siehe Abbildung 6-5).

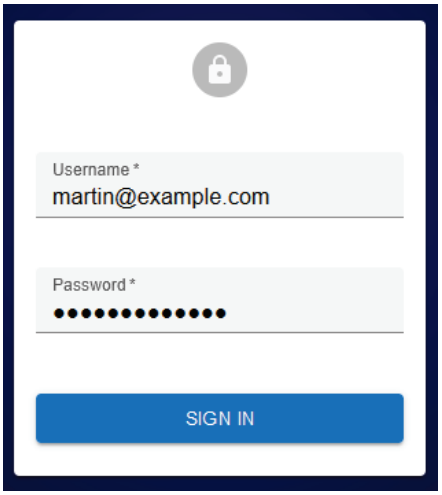

Abbildung 6-5: Anmeldung und Authentifizierung des festgelegten Benutzers (eigene Darstellung)

Nach der erfolgreichen Authentifizierung navigiert *Martin* über das Seitenmenü der Webanwendung zur Sektion *Licences* und sucht dort nach der Lizenz mit der spezifischen ID *X53AfGII7yGARxqYXk9Y*. Er selektiert die entsprechende Lizenz und initiiert eine Zugriffsanfrage, die an den Lizenzvertrauensagenten gesendet wird (siehe Abbildung 6-6).

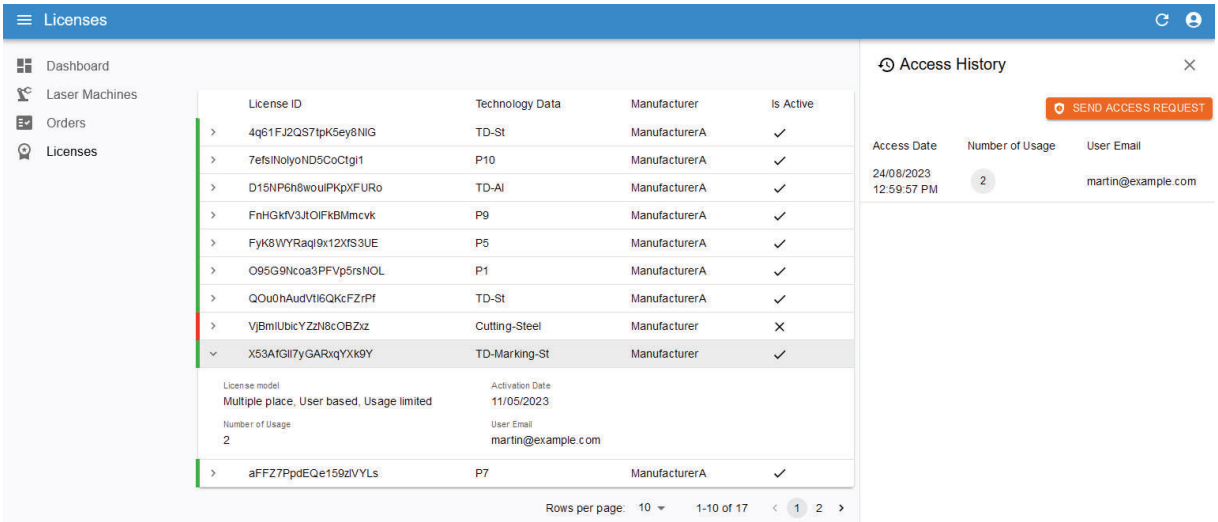

Abbildung 6-6: Auswahl der benötigten Lizenz und Schicken der Zugriffsanfrage (eigene Darstellung)

In den detaillierten Lizenzinformationen ist Martin mit seiner E- Mail- Adresse als autorisierter Benutzer aufgeführt. Zudem werden auf der rechten Seite des Benutzerinterfaces Informationen zur Zugriffshistorie sowie zur verbleibenden Anzahl der zulässigen Nutzungen angezeigt. Diese

Daten bieten eine transparente Übersicht über die bisherige und noch mögliche Nutzung dieser Lizenz.

Nach dem Auswählen der Option *Send access request* wird Martin aufgefordert, sein Passwort erneut einzugeben, um seine Identität zu verifizieren. Zusätzlich erhält er über die in den Lizenzinformationen hinterlegte E- Mail- Adresse einen einmaligen Verifizierungscode, den er im folgenden Schritt eingeben muss. Im Bereich *Maschinenfeld* ist *Martin* darüber hinaus angehalten, die spezifischen Laserbearbeitungsmaschinen des Modells *TruMark Station 3000* auszuwählen, an deren Kryptosysteme die entschlüsselten Technologiedaten weitergeleitet werden sollen (siehe Abbildung 6- 7). Diese Auswahl stellt sicher, dass die Daten ausschließlich an die dafür vorgesehenen Maschinen übermittelt werden, was zur Datensicherheit beiträgt.

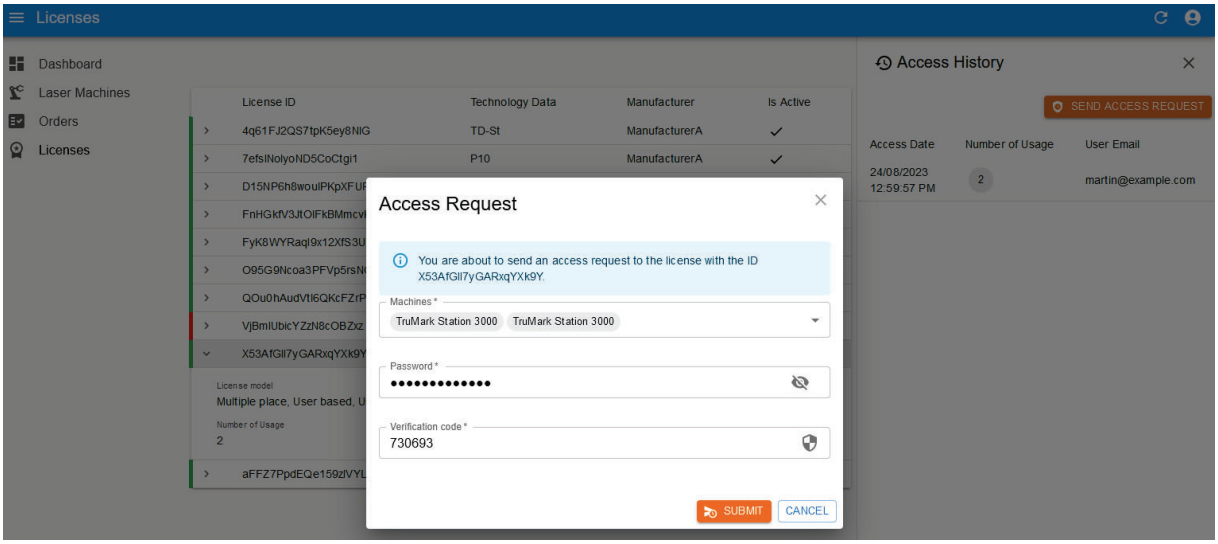

Abbildung 6-7: Benötigte Informationen für die Zugriffsanfrage (eigene Darstellung)

Im Lizenz-Vertrauensagenten werden die Lizenzbedingungen mittels des Smart Contracts evaluiert. Bei einer positiven Überprüfung der Lizenzgültigkeit werden die für die Entschlüsselung notwendigen Daten zurückübermittelt (siehe Abbildung 6-7). Diese Transaktion wird zeitgestempelt und in der Blockchain festgehalten. Gleichzeitig werden die Lizenzbedingungen im Smart Contract aktualisiert, indem die Anzahl der erlaubten Nutzungen um eine reduziert wird. Abschließend werden die empfangenen Entschlüsselungsinformationen im TPM entschlüsselt und im Kryptosystem verwendet, um die gespeicherten Technologiedaten zu entschlüsseln. Nach dem Entschlüsselungsprozess werden die Technologiedaten an die spezifizierten Laserbearbeitungsmaschinen – in diesem Fall jene des Modells *TruMark Station 3000* von *TRUMPF* – übermittelt (siehe Abbildung 6-8).

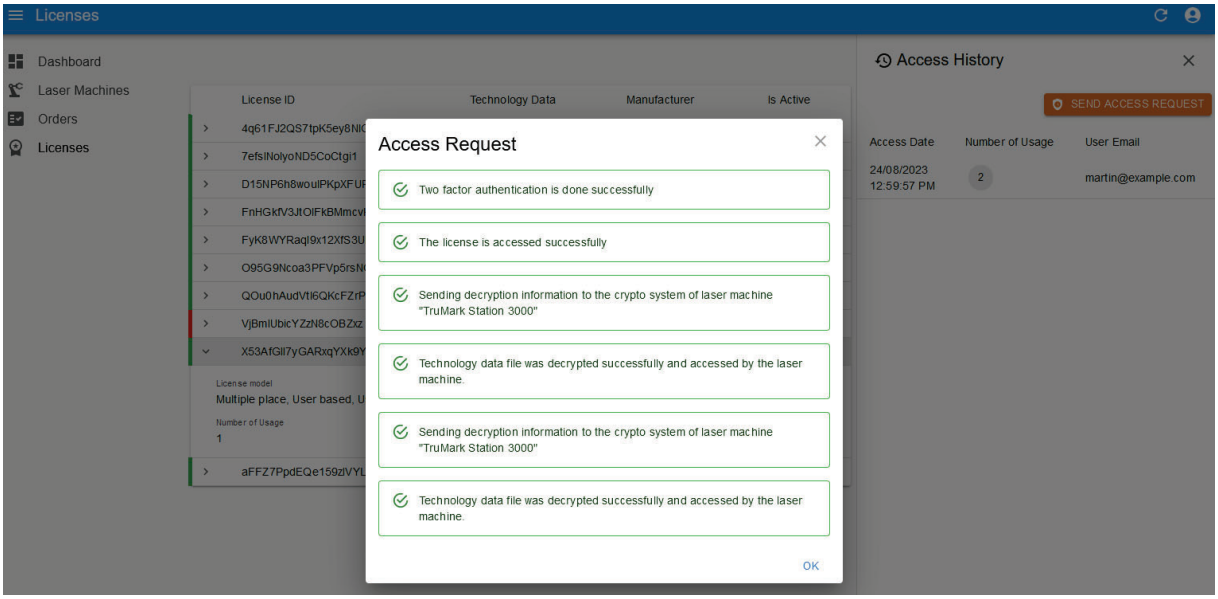

Abbildung 6-8: Rückmeldung zur Erfolg der Lizenzzugriffs und des Erhalts der Technologiedaten (eigene Darstellung)

Die Implementierung einer Lizenzhistorie ermöglicht es dem Anwender, den Verbrauch der Lizenzen genau zu verfolgen. In der angezeigten Übersicht sind für jede Lizenz sowohl die Anzahl der erfolgten Zugriffsanfragen als auch der aktuelle Gültigkeitsstatus ersichtlich. Im betrachteten Anwendungsfall wurde die Lizenz ursprünglich mit einem Kontingent von drei Nutzungen erworben. Nach zwei erfolgreich durchgeführten Zugriffsanfragen wurde dieses Kontingent im Lizenzvertrag entsprechend aktualisiert und auf eine verbleibende Nutzung reduziert. Dies wird deutlich in der Übersicht dargestellt (siehe Abbildung 6-9).

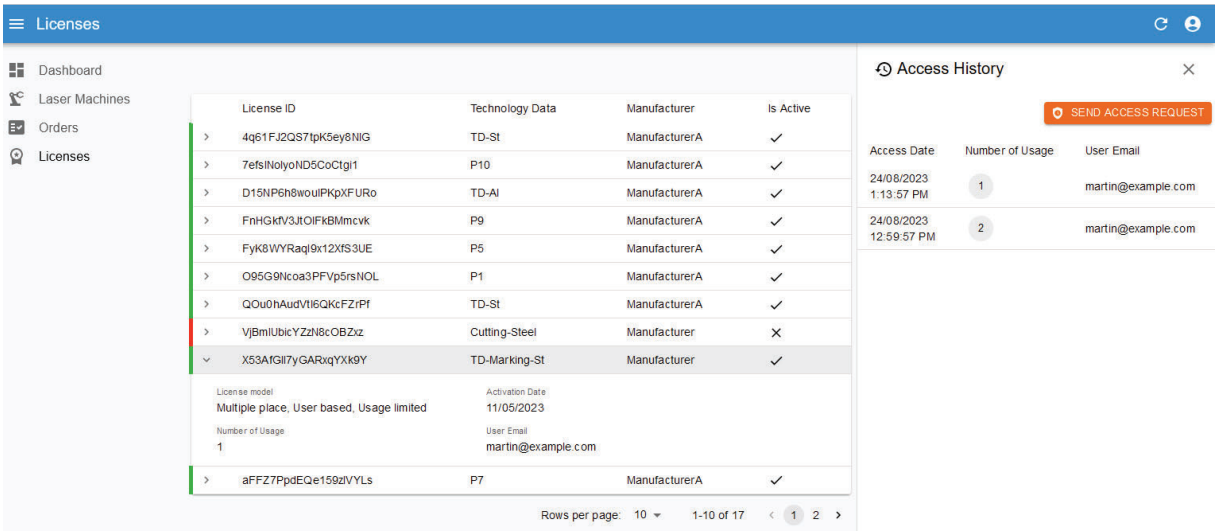

Abbildung 6-9: Übersicht zur Zugriffshistorie und Anzahl der erlaubten Nutzungen (eigene Darstellung)

Wenn ein Benutzer versucht, auf die nutzergebundene Lizenz zuzugreifen, ohne die erforderlichen Kriterien zu erfüllen – in Fall die Übereinstimmung der E- Mail- Adresse und des Passworts –, wird seine Anfrage vom Lizenzvertrauensagenten abgelehnt. Der Benutzer erhält eine Rückmeldung, die den Ablehnungsgrund, etwa eine nicht übereinstimmende E – Mail - Adresse, klar benennt (siehe Abbildung 6-10).

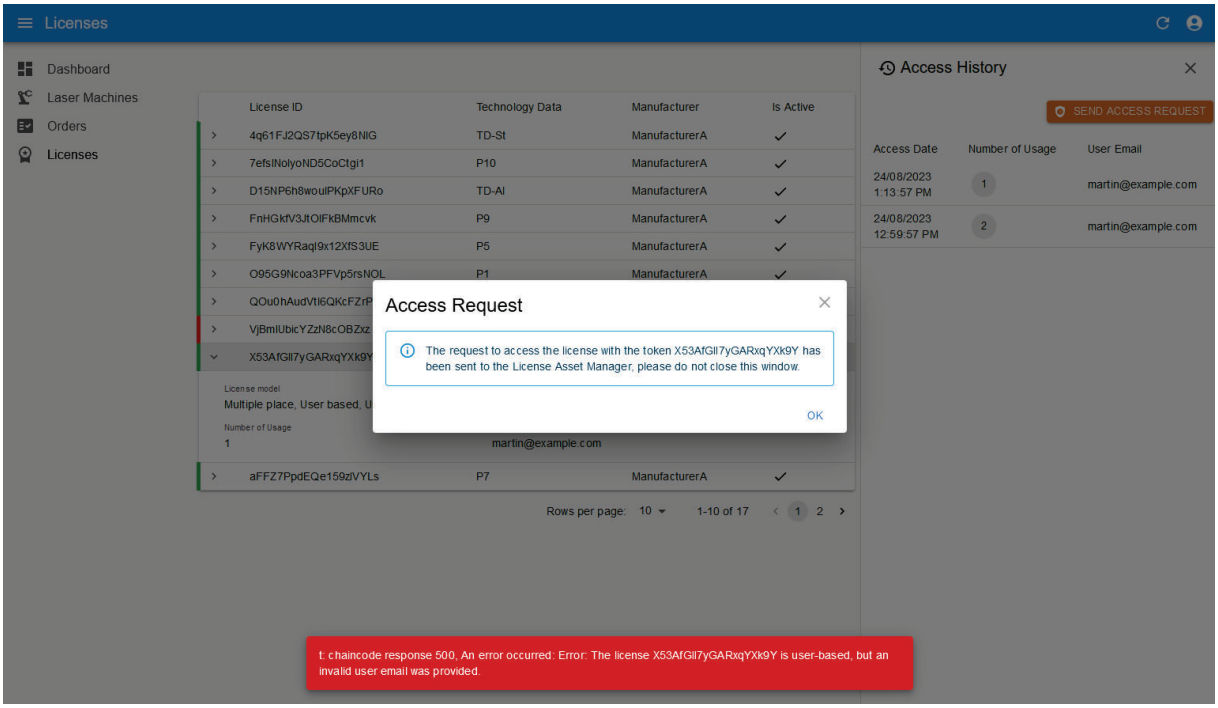

Abbildung 6-10: Fehlermeldung bei unautorisierter Zugriffsanfrage (eigene Darstellung)

6.3 Bewertung der Sicherheit

Die Sicherheit ist eine zentrale Anforderung an das System. Deshalb erfolgt in diesem Unterkapitel eine qualitative Evaluierung der Sicherheitsmerkmale des konzipierten Systems. So viel sei bereits verraten: Die in der vorliegenden Dissertation dargelegten Methoden und Ansätze haben sich als geeignet erwiesen, um die Sicherheitsanforderungen zu erfüllen.

Besonderes Augenmerk gilt hierbei der Komponente des Lizenz-Vertrauensagenten, der eine zentrale Rolle bei der Verwaltung von Lizenzen sowie beim Schutz sensibler Technologiedaten spielt. Zur Implementierung dieses Lizenz-Vertrauensagenten wurde das *Hyperledger Fabric*-Framework verwendet, das bereits breite Anerkennung für seine Sicherheitsmerkmale erhalten hat. Das Sicherheitsframework des *Hyperledger Fabric*-Netzwerks basiert auf einer Vielzahl von kryptografischen Verfahren. Dazu gehören die Integration digitaler Signaturen zur Authentifizierung und Integritätsprüfung, Hash-Funktionen zur Gewährleistung der Unveränderbarkeit von Daten sowie Verschlüsselungsmechanismen zur Wahrung der Datenvertraulichkeit. Diese Merkmale gewährleisten nicht nur die Sicherheit des Netzwerks, sondern auch die der darin verwalteten Lizenzen.

Die Entschlüsselungsinformationen, die ein integraler Bestandteil einer Lizenz sind, spielen eine essenzielle Rolle bei der Entschlüsselung der Technologiedaten. Diese Informationen werden in kryptografisch verschlüsselter Form übertragen und persistent in der Blockchain gespeichert. Ergänzend dazu werden die Technologiedaten dem Endbenutzer (Anwender) in einem verschlüsselten Dateiformat bereitgestellt, um höchste Vertraulichkeit zu gewährleisten. Die Speicherung des privaten Schlüssels des Anwenders auf einem Trusted Platform Module „*OPTIGA™ TPM SLM 9670*“ schafft eine robuste und manipulationssichere Umgebung und reduziert Angriffspotenziale. Die Schlüssellänge orientiert sich an den entsprechenden Empfehlungen des *National Institute of Standards and Technology* (NIST), um ein angemessenes Sicherheitsniveau zu erreichen.

Die Integrität der Technologiedaten und der zugehörigen Lizenzen wird durch die Verwendung von digitalen Signaturen und kryptografischen Hash-Funktionen gewährleistet. Diese Mechanismen dienen dazu, unbemerkte Modifikationen der Daten zu verhindern. Insbesondere wird die digitale Signatur der Entschlüsselungsinformationen in der Blockchain dauerhaft gespeichert und vor ihrer weiteren Nutzung durch das Kryptosystem mittels des öffentlichen Schlüssels des Technologiedatenmarktplatzes einer Verifikation unterzogen. Diese zusätzlichen

Sicherheitsmaßnahmen sorgen dafür, dass die Integrität der Daten während der Netzwerkübertragung unverändert erhalten bleibt. Des Weiteren implementiert das Kryptosystem eine eingehende Prüfung der digitalen Signaturen, die den Technologiedaten anhaften, um deren Authentizität sowie ihre Unversehrtheit sicherzustellen. Diese Überprüfung stellt nicht nur die Herkunft, sondern auch die Integrität der Daten sicher. Die eingesetzten Konsensalgorithmen innerhalb der Blockchain-Infrastruktur tragen ferner dazu bei, dass ein einheitlicher Datenzustand über alle beteiligten Parteien hinweg konsistent gehalten wird. Dies gewährleistet eine zusätzliche Ebene der Datenintegrität und trägt zur Gesamtsicherheit des Systems bei.

Die implementierte Architektur nutzt segmentierte Kommunikationskanäle, die spezifisch für verschiedene Teilnehmergruppen – im gegebenen Kontext ein Anwender und ein Technologiedatenmarktplatz – eingerichtet werden. Diese Segmentierung fungiert als ein Schutzmechanismus gegen unautorisierte Zugriffsversuche auf Lizenzen, indem sie Zugriffsanfragen ausschließlich für vorab definierte Parteien zulässt. Die Authentifizierungsprozesse im System basieren auf anerkannten Identifikationsmechanismen und Zertifikaten, wodurch nur autorisierte Teilnehmer in der Lage sind, sensible Daten zu konsultieren oder Transaktionen innerhalb des Blockchain-Netzwerks auszulösen.

Zur primären Authentifizierung wird ein Verfahren implementiert, das auf der Kombination von Benutzernamen und Passwörtern basiert. Diese erste Ebene der Authentifizierung wird durch eine strenge Passwortrichtlinie gestützt, welche die Generierung sicherer Passwörter fordert und eine Obergrenze für fehlgeschlagene Anmeldeversuche festlegt. Für eine verstärkte Absicherung des Zugriffs auf Lizenzen wird ein Zwei-Faktor-Authentifizierungsmechanismus (2FA) eingeführt. Hierbei ist der Anwender aufgefordert, neben seinem Passwort einen zusätzlichen Verifizierungscode einzugeben, der an die registrierte E- Mail- Adresse gesendet wird. Diese zusätzliche Authentifizierungsschicht minimiert das Risiko von *Account-Takeover-Attacken*, bei denen unbefugte Akteure versuchen könnten, die Anmeldedaten eines legitimen Benutzers zu missbrauchen. Zusätzlich wird ein Authentifizierungstoken eingesetzt, das bei längerer Inaktivität abläuft. Dies dient als weiterer Schutzmechanismus gegen unberechtigten Zugriff und erhöht die Gesamtsicherheit des Systems durch die Implementierung von zeitlich begrenzten Berechtigungen.

Die Implementierung von *gRPC* als Kommunikationsprotokoll für die Interaktion mit dem Lizenz-Vertrauensagenten stellt eine robuste und effiziente Methode für die Übermittlung von Transaktionen innerhalb des Blockchain-Netzwerks dar. Das *gRPC*-Protokoll bietet dabei

mehrere Vorteile, darunter eine starke Verschlüsselung, geringe Latenzzeiten und die Möglichkeit, komplexe Datenstrukturen auf effiziente Weise zu übertragen.

Ergänzend dazu kommt die Verwendung von *RESTful*-APIs ins Spiel, die eine nahtlose Integration mit der Backend-Applikation ermöglichen. Die Sicherheit dieser APIs wird durch eine Reihe von Maßnahmen erhöht: Zum einen erfolgt eine Einschränkung des Zugriffs auf definierte IP-Adressbereiche, sodass nur Anfragen von vorgegebenen IPs akzeptiert und alle anderen automatisch abgelehnt werden. Zum anderen wird eine Basisauthentifizierung implementiert, die eine weitere Sicherheitsschicht für den Zugang zu den Ressourcen bietet.

Das in der vorliegenden Dissertation vorgestellte System gewährleistet durch den Einsatz digitaler Signaturen weiterhin die Nicht-Abstreitbarkeit von Transaktionen. Diese kryptographischen Belege minimieren die Möglichkeit für Parteien, ihre Beteiligung an Transaktionen zu leugnen. Zudem erhöht die Architektur die Systemverfügbarkeit durch in die *Hyperledger Fabric* integrierte Fehlertoleranz- und Redundanzmechanismen. Diese Maßnahmen sorgen dafür, dass das Netzwerk auch bei Ausfall einzelner Knoten funktionsfähig bleibt. In Summe bietet das System ein robustes Framework für eine sichere und kontinuierliche Transaktionsabwicklung im Blockchain-Netzwerk.

Die Verwendung von *Hyperledger Fabric*, einer privaten und genehmigungspflichtigen Blockchain, biete inhärente Schutzmechanismen gegen *Distributed Denial of Service* (DDoS)-Angriffe. In diesem Kontext ist zu beachten, dass Transaktionen ausschließlich von autorisierten Parteien initiiert werden können, die über entsprechend verifizierte Identitäten in Form von Zertifikaten verfügen. Ein Angreifer müsste daher nicht nur Zugang haben zu einem privaten Schlüssel einer solchen autorisierten Identität erhalten, sondern auch die zugriffsbeschränkenden Richtlinien der Plattform umgehen, die Transaktionen auf bestimmte Rollen oder Funktionen beschränken. Die Wahrscheinlichkeit für das erfolgreiche Umgehen dieser Richtlinien ist aufgrund der robusten Sicherheitsarchitektur von *Hyperledger Fabric* als gering einzuschätzen.

Zusätzlich ist die Webapplikation in einer geschützten Netzwerkumgebung hinter einer Firewall positioniert, die als erste Verteidigungslinie gegen Angriffe dient. Dies minimiert das Risiko einer Netzwerküberlastung durch unautorisierte Zugriffsversuche. Darüber hinaus wird die sichere Kommunikation zwischen den verschiedenen Systemkomponenten durch die Implementierung des SSL/TLS-Protokolls gewährleistet. Dieses Verschlüsselungsprotokoll schützt die Datenintegrität und -vertraulichkeit während der Datenübertragung und minimiert

das Risiko von *Man-in-the-Middle*-Angriffen, bei denen ein Angreifer versucht, die Kommunikation abzuhören oder zu manipulieren. Die Einführung des SSL/TLS-Protokolls stellt eine zusätzliche Sicherheitsebene dar, die die Wahrscheinlichkeit einer kompromittierten Kommunikation reduziert.

6.4 Bewertung der Leistungsfähigkeit

Die zweite zentrale Anforderung an das System ist die Leistungsfähigkeit. In diesem Abschnitt erfolgt eine Performance-Evaluation des vorgestellten Systems in Bezug auf kritische Funktionen, darunter der Lizenzzugriff, die Abrufung von Entschlüsselungsinformationen sowie die Entschlüsselung der Technologiedaten. Es werden zwei unterschiedliche Szenarien in Bezug auf die räumliche Verteilung der Systemkomponenten evaluiert. Im ersten Szenario befinden sich alle Komponenten innerhalb des lokalen Netzwerks einer Produktionsstätte. Der Lizenz- Vertrauensagent, der Lizenzmanager und die Backend- Applikation sind alle auf einem einzigen Server konsolidiert, der auf einer *Windows*-Plattform mit einer *Intel(R) Core(TM) i5-1035G1 CPU @ 1.00GHz 1.19 GHz* und 8 GB RAM betrieben wird. Das Kryptosystem, das für den Entschlüsselungsprozess verantwortlich ist, wird auf einem Einplatinencomputer implementiert. Dieses Gerät, ein *Raspberry Pi 4 Modell B*, ist mit einem *Quad-Core Cortex-A72 (ARM v8) 64-bit SoC @ 1.8GHz* und 4 GB RAM ausgestattet [201].

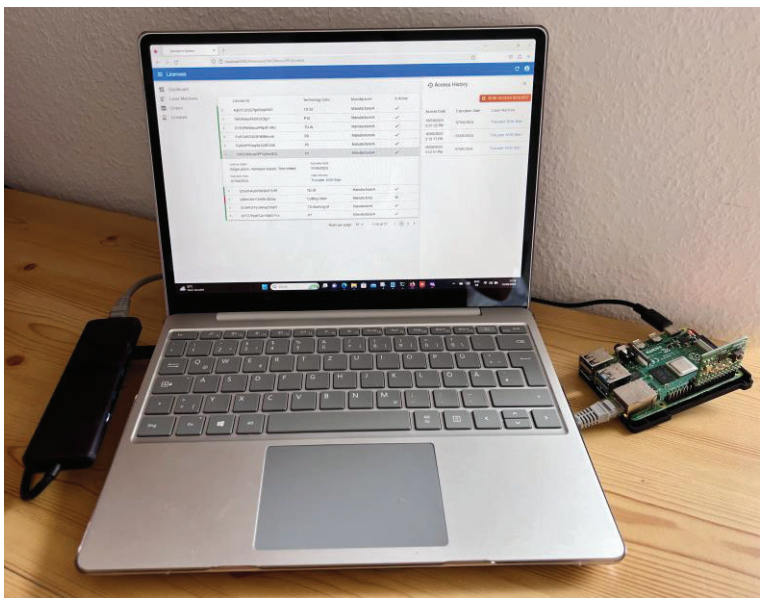

Abbildung 6-11: Aufbau des Demonstrators im ersten Szenario

Im Kontext der Sicherheitsmechanismen wird mit dem *OPTIGA™ TPM SLM 9670* ein TPM an den Einplatinencomputer angeschlossen, um den privaten Schlüssel des Anwenders sicher zu speichern. Die verschlüsselten Technologiedaten und die zugehörigen digitalen Signaturdateien werden in einem dedizierten Ordner innerhalb des lokalen Netzwerks hinterlegt, auf den der Einplatinencomputer Zugriff hat (Siehe Abbildung 6-11).

Im zweiten Szenario wird die Architektur modifiziert, indem der Lizenz-Vertrauensagent in eine externe Cloud-Umgebung verlagert wird. Diese Komponente wird auf einer *Elastic Compute Cloud (EC2) t2.micro*-Instanz von *Amazon Web Services* (AWS) [202] in der Region Frankfurt implementiert. Die restlichen Systemkomponenten, darunter der Lizenzmanager und die Backend-Applikation, verbleiben in der ursprünglichen lokalen Konfiguration.

Zur Leistungsbeurteilung des Systems wurden zehn unterschiedliche Technologiedaten-Dateien mit variablen Größen (von 2,8 KB bis 157.000 KB) vorbereitet und symmetrisch verschlüsselt. Jede dieser Dateien wurde mit einer individuellen Lizenz versehen, die die erforderlichen Entschlüsselungsinformationen in kodierter Form enthält. Diese Lizenzen wurden dem Lizenz- Vertrauensagenten in der Blockchain- Umgebung hinzugefügt und für autorisierte Benutzer freigeschaltet.

Um die Leistungsfähigkeit des implementierten Systems zu evaluieren, wurde der Zeitaufwand ermittelt, die für den vollständigen Zyklus des Datenzugriffs erforderlich ist. Dieser Zyklus umfasst mehrere Schritte: Zwei-Faktor-Authentifizierung des Benutzers, Überprüfung der Lizenzgültigkeit, Erlangen des Zugriffs auf die Lizenz sowie die Entschlüsselung und die Verifizierung der digitalen Signaturen der Technologiedaten. Die Messungen wurden unter den genannten zwei unterschiedlichen Einsatzszenarien durchgeführt: erstens, mit allen Komponenten im lokalen Netzwerk einer Produktionsstätte; und zweitens, mit dem Lizenzvertrauensagenten in einer externen Cloud-Umgebung. Die Zeitmessung begann mit dem Zeitpunkt der Benutzeranfrage für den Zugriff auf eine spezifische Lizenz und endete, sobald die entschlüsselten Technologiedaten an die Laserbearbeitungsmaschine übermittelt worden waren.

Die empirische Analyse wurde für zehn unterschiedliche Technologiedaten-Dateien mit variablen Dateigrößen durchgeführt. Abbildung 6- 12 illustriert die Messergebnisse und zeigt einen klaren Zusammenhang zwischen der Dateigröße und der benötigten Zeit für den gesamten Datenzugriffprozess.

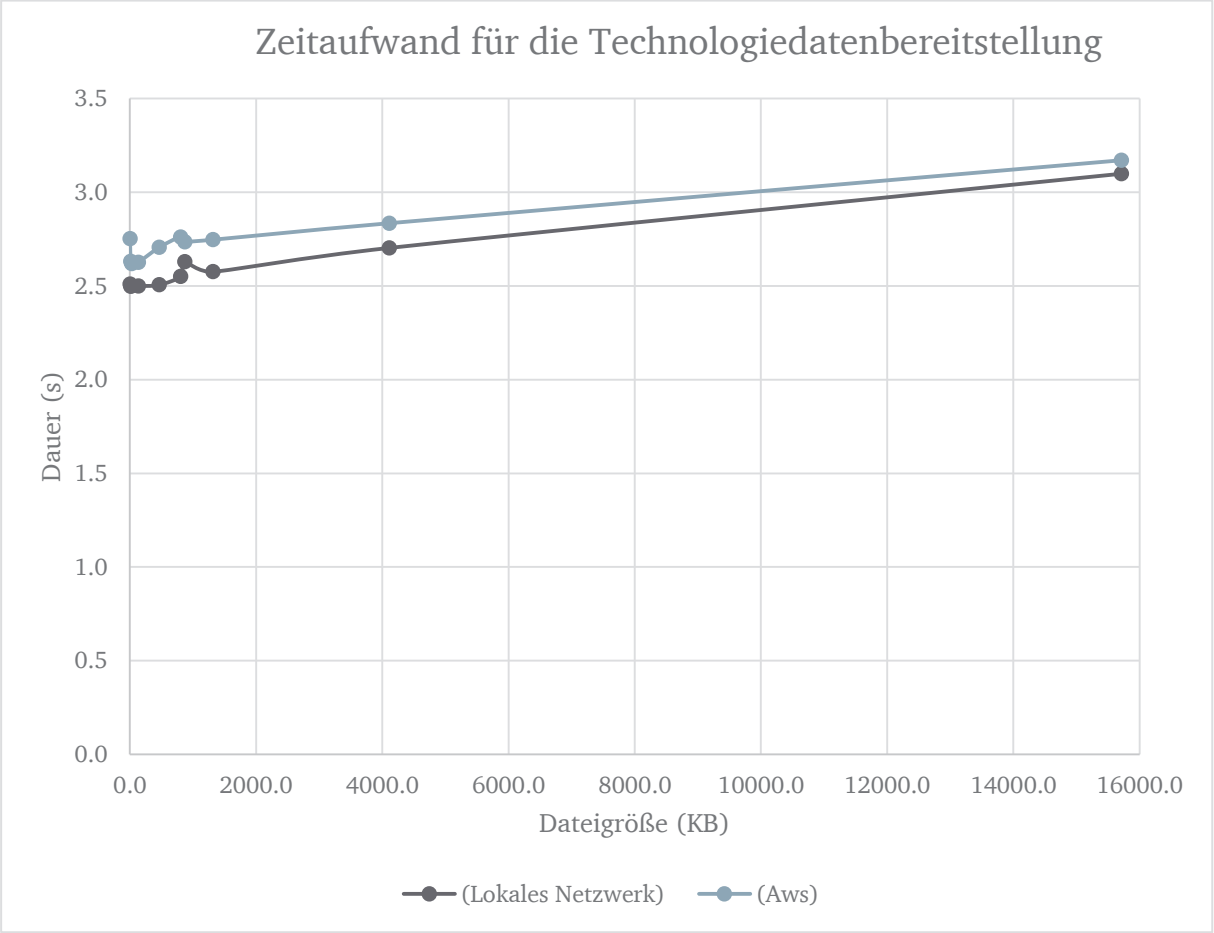

Abbildung 6-12: Messung des Zeitaufwands für die Technologiedatenbereitstellung (eigene Darstellung)

Die Ergebnisse der Leistungsbewertung zeigen, dass die Gesamtdauer für die Datenbereitstellung in beiden konfigurierten Szenarien – lokales und externes Netzwerk – zwischen 2,5 und 3,5 Sekunden variiert. Dies deutet auf eine hohe Effizienz des implementierten Systems hin. Zwei Hauptfaktoren beeinflussen diese Zeitdauer: die Entschlüsselungszeit der Technologiedaten und die Netzwerklatenz.

Die Entschlüsselungszeit hängt im Wesentlichen von zwei Variablen ab: der Dateigröße der zu entschlüsselnden Technologiedaten und der Hardwareleistung des für das Kryptosystem verwendeten Einplatinencomputers. Die Netzwerklatenz wird größtenteils durch die Zugänglichkeit des Lizenz-Vertrauensagenten bestimmt. Dies wird insbesondere im Szenario relevant, in dem der Lizenz-Vertrauensagent auf einer externen *AWS t2.micro*-Instanz gehostet wird.

Die Abbildung 6-13 zeigt den Zeitaufwand für die Datenentschlüsselung und die Signaturprüfung im Kryptosystem, speziell bezogen auf das erste Szenario. In dieser Konfiguration beeinflusst die Performance des Lizenz-Vertrauensagenten das Ergebnis nicht, da alle Systemkomponenten lokal sind. Die Analyse umfasst die Entschlüsselung und Signaturvalidierung der Entschlüsselungsinformationen mit dem *RSA2048*-Algorithmus auf dem *OPTIGA™ TPM SLM 9670-Gerät* und der Technologiedaten mit dem *AES256*-Algorithmus auf dem Einplatinencomputer. Die Entschlüsselungsinformationen sind dabei mit einer Schlüsselgröße von 256 Byte kodiert.

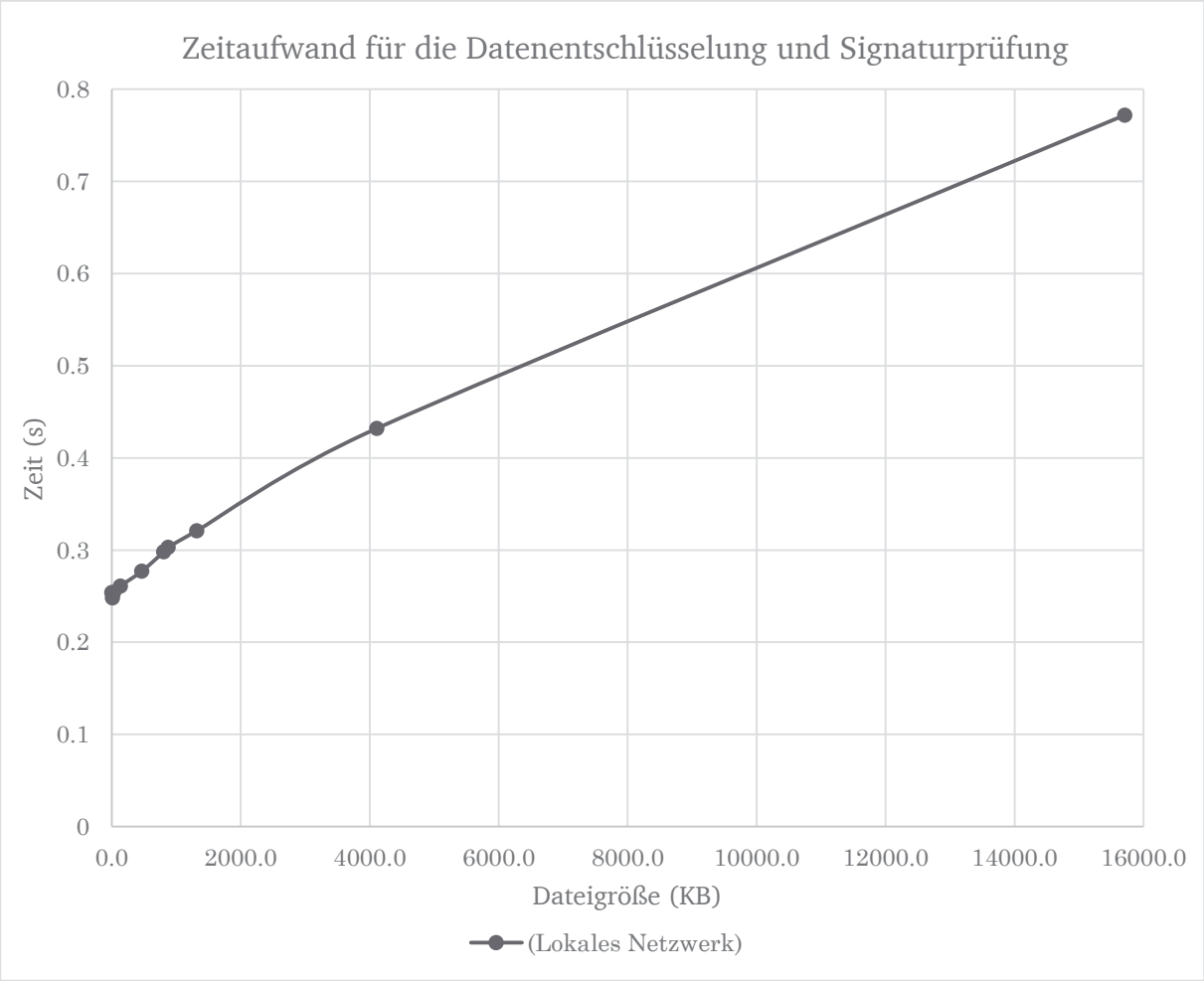

Abbildung 6-13: Messung des Zeitaufwands für die Datenentschlüsselung im Kryptosystem (eigene Darstellung)

Die Abbildung 6-14 verdeutlicht, dass die Zeitspanne für die Datenentschlüsselung und Signaturüberprüfung in Abhängigkeit von der Größe der Technologiedaten zunimmt. Trotz dieser Zunahme bleibt die benötigte Zeit für die Verarbeitung einer Datei mit einer Größe von 15,7 MB unter 0,8 Sekunden. Dieses Ergebnis betont die hohe Effizienz des Kryptosystems,

besondere bei der Verarbeitung größerer Datensätze. Es impliziert, dass die angewendeten Algorithmen und die Systemarchitektur ausreichend leistungsfähig sind, um auch bei steigenden Datengrößen eine hohe Performance zu gewährleisten.

6.5 Verifikation der Anforderungen

Die Tragfähigkeit des entwickelten Konzepts zur Unterstützung des Anwenders bei der effektiven und sicheren Nutzung von Technologiedatenmarktplätzen in der Laserbearbeitung wurde durch die prototypische Implementierung sowie durch die durchgeführte Validierung bereits erfolgreich überprüft. In diesem Unterkapitel soll mit der Verifikation des Konzeptes die Erfüllung der in Kapitel 3 definierten Anforderungen überprüft werden. Der Erfüllungsgrad hinsichtlich der Anforderungen wird in den Tabellen 6-1, 6-2 und 6-4 zusammenfassend dargestellt. Alle Festforderungen wurden vollständig erfüllt.

Die Festforderungen an die Methode zur Konzeption der Unterstützung des Anwenders bei der Nutzung von Technologiedatenmarktplätzen in der Laserbearbeitung sind alle erfüllt. Ein wichtiges Ziel der Konzeption ist die Unterstützung des Anwenders bei der Maschinenauswahl. Die diesbezügliche Methode stützt sich auf einen auftragsspezifischen Ansatz. Dabei spezifizieren die Auftragsdaten wie die Bearbeitungsprozessart, die Bearbeitungstechnologie und die Materialeigenschaften die für die Bearbeitung des Auftrags geeigneten Laserbearbeitungsmaschinen. Die zielführende Methode bei der Konzeption einer rechnerbasierten Lösung besteht in der Verwendung von Entscheidungsbäumen und Mengenoperationen (siehe die Anforderungen 1 und 2). Die Methode dient ebenso der Erreichung des zweiten Ziels, der Unterstützung des Anwenders bei der rechnerbasierten Lizenzmodellauswahl. Hierbei greift die Methode auf Assistenzsysteme zu, um eine effektive Auswahl von Lizenzmodellen basierend auf den Eingaben des Anwenders zu ermöglichen. Ein Inferenzmechanismus greift dabei auf die Wissensbasis sowie auf die festgelegten Auswahlkriterien zu, um die eingegebenen Daten zu verarbeiten und die passenden Lizenzmodelle zu empfehlen (siehe Anforderung 3). Weiterhin ermöglicht die Methode eine gezielte Zuteilung der Daten zu den Laserbearbeitungsmaschinen, indem alle beim Anwender vorhandenen Laserbearbeitungsmaschinen individuelle Identitäten erhalten. So ist der Anwender in der Lage, eine bestimmte Laserbearbeitungsmaschine auszuwählen und die Technologiedaten an diese spezifische Maschine zu schicken (siehe Anforderung 4).

Die Berücksichtigung der Datensicherheit ist bei der Konzeption essentiell. Den Anforderungen an die Authentizität, Vertraulichkeit und Integrität wurde durch die Entwicklung des Sicherheitskonzeptes entsprochen (siehe die Anforderungen 5, 6, und 7). Dabei wurden Technische Schutzmaßnahmen wie die Verschlüsselung von Technologiedaten und deren Entschlüsselungsinformationen zur Sicherstellung der Vertraulichkeit der Daten getroffen. Die Nutzung der digitalen Signaturen gewährleistet die Überprüfung der Datenkorrektheit und -herkunft. Die Methode nutzt weiterhin etablierte Zweifaktor-Authentifizierungslösungen durch die Anforderung von Passwörtern und einmaligen Verifizierungskodes zur Authentifizierung der Benutzer.

Die Nachverfolgung der Technologiedatennutzung ist für die Teilnahme an den Technologiedatenmarktplätzen unerlässlich. Es geht dabei um die Durchsetzung der vereinbarten Nutzungsregeln. Im Rahmen des Lizenzierungskonzepts wurde daher auf deren Überprüfung sowie auf die Nachverfolgung von Änderungen und auf die bedarfsgerechte Verfügbarkeit der Technologiedaten großer Wert gelegt. Die Methode nutzt intelligente Verträge beim Erwerb von Technologiedaten, die die vereinbarten Nutzungsregeln abbilden und automatisch ausführen. Beide Parteien müssen diese Verträge unterzeichnen; dadurch wird die Nicht- Abstreitbarkeit sichergestellt. Die intelligenten Verträge gewährleistet eine kontrollierte Nutzung von Technologiedaten basierend auf Lizenzvereinbarungen, wobei die Lizenzen stets auf ihre Gültigkeit überprüft werden (siehe Anforderung 8 und 11).

Auf der Blockchain werden alle Datenanfragen und Kommunikationsvorgänge aufgezeichnet, die auch nachträglich nicht geändert werden können. Die Methode ermöglicht das Abfragen von Lizenzdaten nach Bedarf und gewährleistet die Sendung der angefragten Entschlüsselungsdaten zur Dechiffrierung der Technologiedaten, wobei die Lizenzdaten nur bei autorisierten Anfragen übermittelt werden. Die Methode erfüllt außerdem die Anforderung an die Verfügbarkeit durch die angemessene Konzipierung von Lösungen zur Datenübertragung und zur Bereitstellung der Technologiedaten (siehe die Anforderung 9 und 10). Die Applikation verfügt über Berechtigungsrichtlinien, wobei Nutzer eingefügt, Rollen definiert und Rechte zugewiesen werden können. Auf der Blockchain werden ebenfalls verschiedene Rollen der Organisationseinheiten bei den Knoten im *Hyperledger Fabric*-Framework definiert (siehe Wunsch 12). Zusammengefasst wurden sowohl Festanforderungen als auch Wünsche an die Methode erfüllt.

Die Festanforderungen an das Informationsmodell sind vollständig erfüllt. Das entwickelte Informationsmodell bildet maschinen-, material-, auftrags- und prozessrelevante Informationen

ab (siehe Anforderung 13). In den verschiedenen Klassendiagrammen werden die Attribute und Methoden sowie Verknüpfungen aller benötigten Informationen zur Unterstützung des Anwenders zur Bereitstellung und Verteilung von Technologiedaten formal abgebildet. Hierbei werden auch die Beziehungen und Verknüpfungen der Datenstrukturen in den verschiedenen Klassendiagrammen beschrieben (siehe die Anforderungen 14 und 15).

Die erforderlichen Informationen zur Auswahl der geeigneten Laserbearbeitungsmaschinen werden ebenfalls abgebildet (siehe Anforderung 16). Hierzu zählt insbesondere, dass der Prozess- und Technologietyp sowie die Materialdaten aus dem Kundenauftrag extrahiert werden. Dabei spielen die Attribute der Laserbearbeitungsmaschinen eine wesentliche Rolle, und sie werden auch abgebildet. Das Informationsmodell beinhaltet die Informationen zur Lizenzmodellauswahl, wobei alle verfügbaren Lizenzmodellattribute, Fakten und Regeln dargestellt werden. Dazu wurden die benötigten Methoden und Attribute abgebildet (siehe Anforderung 17). Die zur kontrollierten Verwendung der Technologiedaten und für die Nutzungsverfolgung erforderlichen Informationen wurden ebenfalls integriert. Hierbei wurden Attribute und Methoden bezüglich der vereinbarten Nutzungsregeln klar zugeordnet und die erforderlichen Beziehungen dargestellt. Das Informationsmodell enthält auch alle für die Überprüfung des Nutzungsstands erforderlichen Technologiedaten (siehe die Anforderungen 18, 19 und 20). Die benötigten Daten zur Überprüfung der Vertraulichkeit, Integrität, Authentizität und Verbindlichkeit der Technologiedaten werden ebenfalls abgebildet. Das Informationsmodell enthält auch alle Attribute und Methoden, die für die Gewährleistung der Datensicherheit gebraucht werden, ebenso wie Schlüsseln und verschlüsselte Daten sowie Methoden zur Entschlüsselung der verschlüsselten Daten und zur Überprüfung der digitalen Signaturen (siehe Anforderung 21).

Die Festforderungen an die Implementierung wurden vollständig erfüllt. Der Anwender interagiert über eine Benutzungsschnittstelle mit der Applikation und greift so auf die entwickelten und implementierten Systemfunktionalitäten zu (siehe Anforderung 22). Das rechnergestützte Werkzeug zur Unterstützung des Nutzers bei der Maschinen- und Lizenzmodellauswahl stellt eine Kernfunktionalität des entwickelten Systems dar. Hierzu wurde ein Assistenzsystem implementiert. Die Applikation ermöglicht dem Anwender die Eingabe der erforderlichen Informationen, die verarbeitet werden, woraufhin nach festgelegten Kriterien und Regeln die passenden Laserbearbeitungsmaschinen und Lizenzmodelle bereitgestellt werden. Diese Handlungsempfehlungen werden dem Anwender über die Benutzerschnittstelle visualisiert und können als CSV-Datei exportiert werden (siehe die Anforderungen 23, 24 und 25).

Die Applikation kann auf einen bereits definierten freigegebenen Ordner zugreifen, um benötigte Technologiedaten bei Bedarf abzurufen. Diese Daten werden dann im Kryptosystem entschlüsselt und an die entsprechende Laserbearbeitungsmaschine weitergeleitet (siehe Anforderung 26). Darüber hinaus bietet die Applikation eine Datenschnittstelle zu dem Lizenz- Vertrauensagenten. Über diese Schnittstelle werden Zugriffsanfragen auf benötigte Lizenzen verschickt, und wenn diese Anfragen berechtigt sind, werden darüber auch die Entschlüsselungsdaten versandt (siehe Anforderung 27). Die zu verarbeitenden Informationen für die Nutzungsverfolgung von Technologiedaten werden der Applikation bereitgestellt. Der Nutzungszustand der Technologiedaten und der zugehörigen Lizenzen wird dann über den Lizenzmanager dem Anwender präsentiert (siehe Anforderung 28). Die Applikation bietet zwei Methoden zur Authentifizierung des Benutzers. Bein Einloggen ist der Benutzer aufgefordert, seine Login-Daten einzugeben. Zusätzlich muss der Anwender bei Zugriffsanfragen auf Lizenzen einen einmaligen Verifizierungscode eingeben. Dies ermöglicht eine Zweifaktor-Authentifizierung und dient der Verbesserung der Datensicherheit (siehe Anforderung 29).

Die Verifikation hat gezeigt, dass alle Festanforderungen an die Methode, das Informationsmodell und die Implementierung erfüllt sind. In den folgenden Tabellen 6-1, 6-2 und 6-3 wird eine Übersicht über die Anforderungserfüllung gegeben.

Tabelle 6-1: Übersicht über die Erfüllung der Anforderungen bezüglich der Methode

Nr.	Bezeichnung	Erfüllungsgrad
Anforderungen an die Methode		
1	Die Methode muss eine Vorgehensweise definieren, anhand der ein auftragsspezifischer Ansatz zur Auswahl der geeigneten Laserbearbeitungsmaschinen entwickelt werden kann.	●
2	Die Methode muss eine Vorgehensweise definieren, anhand der eine rechnerbasierte Maschinenauswahl ermöglicht wird.	●

3	Die Methode muss eine Vorgehensweise definieren, anhand der ein effektiver und rechnerbasierter Ansatz zur Auswahl der passenden Lizenzmodelle realisiert werden kann.	●
4	Die Methode muss eine Vorgehensweise definieren, anhand der eine gezielte Zuteilung der Daten zu den Laserbearbeitungsmaschinen möglich ist.	●
5	Die Methode muss eine Vorgehensweise definieren, anhand der die Vertraulichkeit der verarbeitenden Daten gewährleistet werden kann.	●
6	Die Methode muss eine Vorgehensweise definieren, anhand der die Integrität der Daten sichergestellt wird.	●
7	Die Methode muss eine Vorgehensweise definieren, welche die Authentifizierung der Kommunikationspartner ermöglicht.	●
8	Die Methode muss eine Vorgehensweise definieren, anhand der eine Überprüfung der Nutzungsregeln vor und nach dem Laserbearbeitungsprozess sichergestellt wird.	●
9	Die Methode soll eine Vorgehensweise definieren, die die Nachverfolgung von Änderungen während der Datenverarbeitung ermöglicht.	●
10	Die Methode muss eine Vorgehensweise definieren, anhand der die Verfügbarkeit von Daten sichergestellt wird.	●

11	Die Methode muss eine Vorgehensweise definieren, anhand der eine Überprüfung der Verbindlichkeit bzw. Nicht-Abstreitbarkeit sichergestellt wird.	●
12	Die Methode muss eine Vorgehensweise definieren, anhand der eine Autorisierungsstrategie zur Verwaltung der Zugriffsrechte von Nutzern erarbeitet wird.	●

Tabelle 6-2: Übersicht über die Erfüllung der Anforderungen bezüglich des Informationsmodells

Nr.	**Bezeichnung**	**Erfüllungsgrad**
Anforderungen an das Informationsmodell		
13	Das Informationsmodell muss maschinen-, material-, auftrags- und prozessrelevante Informationen enthalten.	●
14	Das Informationsmodell muss die Datenstrukturen und -beziehungen formal abbilden.	●
15	Das Informationsmodell muss die Verknüpfung zwischen den Datenstrukturen übersichtlich repräsentieren.	●

16	Das Informationsmodell muss für die Auswahl der geeigneten Laserbearbeitungsmaschinen die erforderlichen Informationen und Beziehungsstrukturen abbilden.	●
17	Das Informationsmodell muss die für die Lizenzmodellauswahl erforderlichen Informationen und Beziehungsstrukturen abbilden.	●
18	Das Informationsmodell muss Informationen aus der Nutzungsphase bezüglich der vereinbarten Nutzungsregeln von erworbenen Technologiedaten integrieren und bereitstellen.	●
19	Das Informationsmodell muss Informationen zur Prüfung der Nutzungsregeln abbilden und zur Verfügung stellen.	●
20	Das Informationsmodell muss eine klare Zuordnung der Lizenzen und Lizenzmodellen zu den Technologiedaten ermöglichen.	●
21	Das Informationsmodell muss Informationen zur Prüfung der Datensicherheit abbilden und zur Verfügung stellen.	

Tabelle 6-3: Übersicht über die Erfüllung der Anforderungen bezüglich der Implementierung

Nr.	**Bezeichnung**	**Erfüllungsgrad**
Anforderungen an die Implementierung		
22	Die Applikation muss dem Anwender eine grafische Benutzungsoberfläche zur Interaktion und Navigation bereitstellen.	●

23	Die Applikation muss die Durchführung der Maschinenauswahl ermöglichen.	●
24	Die Applikation muss die Eingabe der für die Lizenzmodellauswahl nötigen Informationen ermöglichen.	●
25	Die Applikation muss die Empfehlungen zur Maschinen- und Lizenzmodellauswahl bereitstellen.	●
26	Die Applikation muss eine Datenschnittstelle für das Abziehen der Technologiedaten und Weiterleiten an die Laserbearbeitungsmaschine bereitstellen.	●
27	Die Applikation muss eine Datenschnittstelle für die Bereitstellung der Lizenzen zur Verfügung stellen.	●
28	Die Applikation muss die Informationen zum Nutzungszustand darstellen können.	●
29	Die Applikation muss die Authentifizierung des Benutzers ermöglichen.	●

6.6 Fazit

Basierend auf der Implementierung der Webapplikation, des Lizenz- Vertrauensagenten und des Kryptosystems erfolgte in diesem Kapitel die Validierung und Verifizierung des in der vorliegenden Dissertation entwickelten Konzepts. Methodisch stützte sich die Validierung auf den *Cognitive Walkthrough*, wobei ein repräsentativer Anwendungsfall zum Lasermarkierprozess zur Überprüfung des Systems herangezogen wurde. Dieser Anwendungsfall wurde sorgfältig ausgewählt, um die Vielseitigkeit des Systems zu prüfen und

die erwartete Gebrauchsweise des Systems möglichst vollständig abzubilden. Der Anwendungsfall simuliert realistische Kundenaufträge und demonstriert, dass alle Funktionen der Applikation wie geplant ablaufen. Durch die Implementierung der Webapplikation, des Lizenz- Vertrauensagenten in einer Blockchain-Umgebung und des Kryptosystems wurde zudem ein hohes Maß an Datensicherheit erreicht.

Das System zeigte dabei seine Fähigkeit, den Anwender durch automatisierte Empfehlungen in der Maschinen- und Lizenzauswahl zu unterstützen. Die Empfehlungen werden in Form von CSV-Dateien exportiert und können daher direkt für die Auftragsplanung verwendet werden. So werden die Nutzer dabei unterstützt, die für den jeweiligen Auftrag am besten passende Laserbearbeitungsmaschine und das geeignetste Lizenzmodell auszuwählen. Das System ist flexibel und bietet dem Anwender ein hohes Maß an Kontrolle und Personalisierung, da die Benutzereinstellungen veränderbar sind und der Nutzer zur Eingabe der erforderlichen Informationen im Rahmen der Beantwortung eines Fragenkatalogs aus vorgegebenen Antwortmöglichkeiten wählen kann. Des Weiteren bietet die Applikation eine klare Übersicht über verfügbare Laserbearbeitungsmaschinen als Auswahlliste, was den Lizenzierungsprozess erleichtert und Fehler minimiert.

Die Validierung hat gezeigt, dass das implementierte System für die Realisierung des entwickelten Sicherheitskonzepts geeignet ist. Die Datensicherheit wird durch eine Vielzahl an Mechanismen, einschließlich der Integration eines Lizenzvertrauensagenten in eine Blockchain-Umgebung, gewährleistet. Die Implementierung des Lizenz- Vertrauensagenten in die Blockchain-Umgebung ermöglicht eine manipulationssichere Lizenzverwaltung und minimiert das Risiko externer Angriffe.

Die private Natur des *Hyperledger Fabric*-Frameworks und die Verwendung von vordefinierten Kanälen für bestimmte Parteien beseitigt die Gefahr der Einflussnahme durch Außenstehende vollständig. Nur autorisierte Parteien können Anfragen zu Lizenzen stellen und entsprechende Transaktionen durchführen. Die integrierten Sicherheitsmechanismen, einschließlich der Zwei-Faktor-Authentifizierung, trugen wesentlich zur Manipulations- und Datensicherheit bei. Die Eingaben der potentiellen Nutzer (Benutzername und Passwort) werden zunächst verifiziert, bevor eine Transaktion genehmigt wird. Im Anwendungsfall wurde gezeigt, dass die Zugriffanfrage auf eine Lizenz bei fehlender Authentifizierung abgelehnt wird. Im Sinne der Transparenz gibt der Lizenzmanager dem Anwender zudem eine Übersicht über die erworbenen Lizenzen und deren Zustand inklusive weiterer nützlicher Informationen. Der Einsatz eines

Raspberry Pi 4 Model B und eines TPM stellt darüber hinaus eine kosteneffektive und sichere Lösung dar für die Verwaltung von kryptografischen Schlüsseln und sensiblen Daten.

Zusätzlich wurden die Sicherheit und die Leistungsfähigkeit des Systems tiefgehend untersucht. Bei der Untersuchung der Sicherheit wurde festgestellt, dass die Schutzziele bei der Konzipierung und Implementierung des Systems berücksichtigt und erfüllt wurden. Die Verwendung von neuesten Technologien wie Blockchain und Smart Contract ermöglichen die sichere Lizenzierung von Technologiedaten und deren kontrollierte Nutzung. Die Authentifizierung von beteiligten Parteien minimiert das Potential von Angriffen und schützt daher die Lizenzen vor Manipulation oder unautorisierter Nutzung.

Die Systemleistung wurde unter realitätsnahen Bedingungen getestet und zeigte eine hohe Performanz in Bezug auf die Datenlieferung. Dabei wurden zwei Szenarien mit zwei unterschiedlichen Orten für den Lizenz-Vertrauensagenten bzw. für das Blockchain-Netzwerk untersucht. Die Konfiguration der Szenarien bietet einen Einblick in die Systemleistung unter kontrollierten, aber dennoch realitätsnahen Bedingungen, und ermöglicht eine Beurteilung der Effizienz des Systems für spezifische industrielle Anwendungen. Diese zwei Szenarien erlauben also eine differenzierte Analyse der Systemleistung unter variierenden Betriebsbedingungen. Insbesondere wird die Auswirkung der Entkopplung des Lizenz- Vertrauensagenten auf die Gesamtperformanz des Systems evaluiert. Dabei ermöglicht die *AWS*-Cloud-Instanz eine hohe Skalierbarkeit und Flexibilität, während das lokale Szenario den Fokus auf ein kontrolliertes und spezifisches Umfeld legt. Die Vielzahl an verwendeten Dateigrößen ermöglicht eine umfassende Analyse der Systemeffizienz und -latenz, insbesondere in Bezug auf das Abrufen der Entschlüsselungsinformationen aus der Blockchain und die nachfolgende Entschlüsselung der Technologiedaten. So konnte der Einfluss der Dateigröße auf die Gesamteffizienz des Systems systematisch erfasst werden. In beiden Szenarien lag die Gesamtzeit für die Datenlieferung beginnend mit der Lizenzanfrage bis zur Übermittlung der Technologiedaten an die Maschine unter drei Sekunden. In dieser Zeitspanne wurde auch die Authentifizierung, Entschlüsselung und Signaturüberprüfung vollzogen. Diese systematische Evaluierung ermöglichte ein tieferes Verständnis der Skalierbarkeit und Effizienz des implementierten Systems und bestätigte seine Eignung für Echtzeitanwendungen in industriellen Kontexten. Zusammenfassend haben die tiefgehenden Untersuchungen der Systemsicherheit und -leistungsfähigkeit ergeben, dass die Schutzziele adäquat berücksichtigt wurden und dass der Einsatz von Blockchain- und Smart-Contracts-Technologie eine sichere und transparente Lizenzierung und Nutzung von Technologiedaten gewährleistet. Die Untersuchung bestätigte

zudem eine hohe Effizienz und Skalierbarkeit des Systems, insbesondere in Bezug auf die Datendurchsatzrate und die Latenzzeiten.

Im Anschluss an die Validierung wurde eine Verifikation durchgeführt, mit der die Übereinstimmung zwischen den Eigenschaften des implementierten Systems und den an sie gestellten Anforderungen formal überprüft wurde (siehe Kapitel 3). Die Verifikation hat gezeigt, dass allen Festanforderungen an die Methode, das Informationsmodell und die Implementierung vollständig entsprochen wurde. Mit der erfolgreich durchgeführten Validierung und Verifikation konnte in diesem Kapitel die Tragfähigkeit des entwickelten Konzeptes zur Unterstützung des Anwenders bei der sicheren und effektiven Nutzung von Technologiedatenmarktplätzen in der Lasermarkierung nachgewiesen werden.

7 AUSBLICK

Im Rahmen dieser Dissertation wurde ein innovatives System zur Unterstützung des Anwenders bei der effektiven und sicheren Nutzung von Technologiedatenmarktplätzen am Beispiel der Lasermarkierung entwickelt und evaluiert. Das konzipierte System wurde durch die Entwicklung einer Webapplikation, eines Kryptosystems und eines Lizenz-Vertrauensagenten mittels Blockchain- und Smart-Contract-Technologien prototypisch implementiert. Mithilfe von einem repräsentativen Anwendungsfall für realistische Kundenaufträge konnte die Konzeption auf ihre Tragfähigkeit und Anwendbarkeit überprüft werden. Dabei wurde deutlich, dass das entwickelte System funktioniert und dass es die Vertraulichkeit, Integrität und stete Verfügbarkeit der Technologiedaten gewährleistet. Außerdem bietet das System Unterstützung bei der effektiven Bereitstellung von Technologiedaten, indem es dem Anwender Handlungsempfehlungen zur Maschinen- und Lizenzauswahl zur Verfügung stellt.

In diesem Kapitel werden nun weiterführende wissenschaftliche Fragestellungen und Entwicklungspotentiale dargelegt. Das entwickelte System hat gezeigt, dass eine sichere Lizenzierung von digitalen Technologiedaten in der Lasermarkierung möglich ist, wobei diese Daten nach festen Bedingungen kontrolliert verwendet werden. Ein großer Vorteil hierbei ist die Ermöglichung der Entwicklung neuer innovativer Geschäftsmodelle basierend auf digitale Daten, die bereits in Unternehmen vorhanden sind. Daraus ergibt sich ein zusätzliches Wertschöpfungspotential für die Industrie. Mit dem rapiden Anstieg der Digitalisierung und Vernetzung, insbesondere im industriellen Umfeld, wird vermutet, dass Datenmarktplätze für den Handel mit Technologiedaten oder ähnlichen Produkten zukünftig weiter verbreitet sein werden. Eine interessante Forschungsfrage bezüglich der Erweiterung des Anwendungsspektrums des im Rahmen dieser Dissertation entwickelten Systems lautet, ob es auch für andere Fertigungsprozesse oder sogar in ganz anderen Branchen eingesetzt werden könnte. Und falls diese Frage bejaht werden kann: Welche Anpassungen wären dafür nötig?

Eine Einschränkung des konzipierten Systems besteht darin, dass Anwender die Lizenzen nicht eigenständig verändern können; sie müssen aus vordefinierten Lizenzmodellen wählen. Die Bestimmung der verfügbaren Lizenzmodelle obliegt den Technologiedatenmarktplätzen, die als Lizenzgeber agieren. In dem vorgestellten System wurden vier solcher Lizenzmodelle mittels intelligenter Verträge implementiert.

Das in dieser Dissertation vorgestellte Datenmodell, das Lizenzen auf der Blockchain abbildet, enthält sämtliche Informationen, die zur Überprüfung der zugehörigen Nutzungsbestimmungen erforderlich sind. Diese Nutzungsbestimmungen sind mit einem der vordefinierten Lizenzmodelle verknüpft und einem spezifischen Anwendungsfall zugeordnet. Eine relevante Forschungsfrage könnte in diesem Zusammenhang lauten: Wie könnte eine flexible und individuelle Datenlizenzierung vonstattengehen? Insbesondere wäre zu untersuchen, wie ein Mechanismus aussehen könnte, der es den Anwendern ermöglicht, Lizenzen individuell anzupassen, ohne die Sicherheit oder Integrität der intelligenten Verträge zu beeinträchtigen. Hier wäre eine detaillierte Analyse der Programmlogik des intelligenten Vertrages sowie des Lizenzdatenmodells erforderlich, um herauszufinden, wie solche Änderungen sicher implementiert werden können.

Dazu könnte eine Analyse des Arbeitsaufwands für die Änderung der existierenden Lizenzbedingungen und der dafür notwendigen Arbeitsschritte durchgeführt werden. Der Prozess könnte so gestaltet sein, dass der Anwender zunächst eine Anfrage zur Änderung des Lizenzmodells für eine spezifische Lizenz stellt und dem Lizenzgeber die benötigten Lizenzbedingungen sowie weitere erforderliche Informationen mitteilt. Anschließend könnte der Lizenzgeber den Anwender über eventuell anfallende Zusatzkosten für die gewünschten Änderungen informieren. Stimmt der Anwender zu, interagiert der Technologiedatenmarktplatz mit dem bereitgestellten Smart Contract. Dieser wird dann auf dem entsprechenden Kanal in der Blockchain des Lizenz-Vertrauensagenten veröffentlicht und übermittelt eine Transaktion, die die Lizenzinformationen gemäß den neuen Anforderungen des Anwenders aktualisiert. Im Zusammenhang mit der Skalierbarkeit, Sicherheit und rechtlichen Durchsetzbarkeit bestehen jedoch noch gravierende Herausforderungen beim Einsatz von intelligenten Verträgen. In diesem Kontext ergeben sich noch zahlreiche weitere interessante Forschungsfragen, deren Bearbeitung überaus lohnenswert wäre.

Das im Rahmen dieser Dissertation entwickelte System wurde bislang nur für eine begrenzte Anzahl an Nutzern und Laserbearbeitungsmaschinen implementiert und getestet. Eine weitere Forschungsfrage könnte sich der Skalierbarkeit des Systems widmen. Interessante wäre insbesondere die Überprüfung der Eignung privater Blockchain-Netzwerke für industrielle Anwendungsszenarien. Wie kann das System so entwickelt werden, dass es problemlos eine große Anzahl von Nutzern und eine größere Datenmenge bewältigen kann, ohne die Sicherheit zu beeinträchtigen? Dies könnte die Erforschung weiterer Typen von Netzwerken hinsichtlich ihrer Sicherheit und Leistungsfähigkeit beinhalten.

Durch ein integriertes Assistenzsystem erhalten Anwender automatisch Empfehlungen für die Auswahl von Lizenzmodellen. Die Wissensregeln wurden auf Basis der in dieser Dissertation durchgeführten Recherche festgelegt. Diese könnten durch erfahrene Anwender und Marktanalysen optimiert werden. Zusätzlich könnte die Integration weiterer KI- Technologien ein spannender Forschungsansatz sein [228], [229]. Welche Algorithmen sind herfür am besten geeignet und wie könnte ein Algorithmus trainiert werden, um noch präzisere und individuellere Handlungsempfehlungen für den Anwender bereitzustellen? Dies setzt allerdings die Analyse großer Mengen von historischen Daten und Benutzerinteraktionen voraus, um effektive Lizenzmodelle zu trainieren. Deshalb kann diese Fragestellung für die Anwendung in einer echten Produktionsumgebung interessant sein und ein großes Optimierungspotential entfalten. Ferner könnten Mechanismen zur automatisierten Anpassung von Lizenzbedingungen auf Grundlage der Nutzungsdaten oder spezieller Anforderungen des Anwenders in die Programmlogik des intelligenten Vertrages integriert werden. Dies wäre eine Erweiterung des aktuellen Systems und könnte die Effizienz erheblich steigern.

In Bezug auf die Sicherheit ergeben sich weitere Forschungsfragen: Wie können die neuesten Fortschritte in der Kryptografie genutzt werden, um das Sicherheitsniveau des Systems noch weiter zu erhöhen? Wie können Datenschutz und Sicherheit gewährleistet werden, ohne dabei negative Auswirkungen sowohl auf die Performance als auch auf die Zuverlässigkeit des Systems zu riskieren?

Die Entschlüsselungszeit der Technologiedaten wird sowohl von ihrer Größe als auch von der Hardware-Spezifikation des Einplatinencomputers beeinflusst. Das *Raspberry Pi 4 Modell B* hat sich in den Tests als ausreichend erwiesen, um den Anforderungen gerecht zu werden. Die Systemleistung kann weiter verbessert und die Reaktionszeit verkürzt werden, indem das Kryptosystem in einem leistungsstärkeren Rechner eingerichtet wird. Zudem kam im Rahmen der Validierung des Konzeptes die Frage auf, wie die Webapplikation und der Lizenz- Vertrauensagent in einer realen Industrieumgebung die Benutzer und ihre Rechte verwalten könnten. Es ist eine organisatorische Frage nichts desto trotz kann sie Gegenstand einer wissenschaftlichen Analyse werden. Es wäre beispielsweise denkbar, die Rechte in Bezug auf individuelle Lizenzierungsprozesse näher zu untersuchen und zu bestimmen.

8 ZUSAMMENFASSUNG

Durch die Digitalisierung in der Industrie, die Informationsökonomie und die Industrie 4.0 eröffnen sich wirtschaftliche Potenziale zur Entwicklung neuer Geschäftsmodelle, die auf digitalen Daten basieren. Darüber hinaus bietet die Laserbearbeitung als flexible und innovative Technologie vielfältige Chancen für die Fertigungsindustrie.

In dieser Dissertation wird ein innovatives Geschäftsmodell für den Vertrieb von Technologiedaten in der Lasermarkierung beschrieben: der sogenannte Technologiedatenmarktplatz. Die Nutzung solcher Technologiedatenmarktplätze bringt einige Herausforderungen mit sich, insbesondere in Bezug auf die effektive Bereitstellung von Technologiedaten und die Informationssicherheit. Diese Herausforderungen hindern die Nutzer der Technologiedatenmarktplätze daran, von deren wesentlichem Vorteil – einer potentiellen Reduktion der Produktionskosten in der Laserbearbeitung – zu profitieren.

Das in dieser Dissertation vorgestellte Konzept bewältigt diese Herausforderungen. Denn es wird eine sichere Integration der Laserbearbeitungsmaschinen der Anwender auf den Technologiedatenmarktplätzen ermöglicht, indem die Nutzer beim Bezug und bei der Verwendung lizenzierter Technologiedaten unterstützt werden. Dafür wurde ein Assistenzsystem zur Unterstützung des Anwenders bei der effektiven und sicheren Nutzung von Technologiedatenmarktplätzen speziell für die Lasermarkierung entwickelt und evaluiert. Dieses Assistenzsystem ist dafür geeignet, das komplexe Zusammenspiel einer großen Anzahl an Laserbearbeitungsmaschinen und großer Mengen erworbener Technologiedaten und Lizenzen zu managen.

Das Assistenzsystem hat drei Hauptkomponenten: Maschinenauswahl, Lizenzmodellauswahl und Datensicherheit. Das erstgenannte Konzept stellt Handlungsempfehlungen für die Auswahl von Laserbearbeitungsmaschinen bereit. Es basiert auf einer gründlichen Analyse von Auftragsinformationen sowie der Merkmale der verfügbaren Maschinen. Das Hauptziel besteht darin, dem Anwender bzw. der Auftragsplanung eine fundierte Grundlage für die Auswahl der optimalen Maschine zur Bearbeitung eines spezifischen Auftrags zu bieten. Die zu diesem Zweck getroffene Maschinenvorauswahl der im Rahmen dieser Dissertation entwickelten Applikation geht zurück auf vordefinierte Kriterien, die in das System integriert sind.

Das Systemelement für die Auswahl von Lizenzmodellen wurde als ein wissensbasiertes Assistenzsystem konzipiert, das auf KI-Technologie basiert. Auf Grundlage individueller Auftragsanforderungen und der verfügbaren Lizenzoptionen werden im Rahmen einer rechnergestützten Analyse fundierte Empfehlungen für die Auswahl eines geeigneten Lizenzmodells formuliert. Dabei ist das System flexibel, denn die vordefinierten Benutzereinstellungen sowie die Angaben bezüglich des Auftrags und der vorhandenen Ressourcen, auf deren Grundlage automatisierte Empfehlungen ausgegeben werden, können an die aktuellen betrieblichen Begebenheit der Anwender angepasst werden. Diese Empfehlungen für die Auswahl von Maschinen und Lizenzmodellen tragen zur Optimierung der Produktionsplanung bei und unterstützen eine effiziente Nutzung der Ressourcen.

Der Sicherheitsaspekt ist von immenser Bedeutung bei der Bereitstellung, Verteilung und Nutzung von Technologiedaten. Das entwickelte Konzept gewährleistet einen umfassenden Schutz der Technologiedaten und Lizenzinformationen. Ferner ermöglicht es eine kontrollierte Anwendung der Technologiedaten gemäß den festgelegten Lizenzvereinbarungen. Besonders wichtig ist dabei das Lizenzierungskonzept, welches die konsequente Einhaltung der Lizenzvorgaben sicherstellt und somit das Risiko unbefugter Zugriffe oder Missbräuche von Technologiedaten minimiert. Die Integration von Smart Contracts in eine Blockchain-Technologie trägt zusätzlich zur Erhöhung von Sicherheit und Transparenz bei. Außerdem gewährleistet die Nutzung von etablierten kryptographischen Maßnahmen wie Verschlüsselung und digitale Signaturen gemäß den aktuellen Empfehlungen das Erreichen der Schutzziele in Bezug auf die Informationssicherheit.

Das konzipierte System wurde durch die prototypische Umsetzung einer Webapplikation, eines Kryptosystems und eines Lizenz-Vertrauensagenten im Blockchain-Netzwerk implementiert. Hierbei wurde eine Mikroservice-Architektur gewählt, um sowohl Flexibilität als auch Skalierbarkeit sicherzustellen. Die Umsetzung der Sicherheitsmaßnahmen im Kontext der Bereitstellung der Daten und die Entwicklung entsprechender Mechanismen erfolgte mit besonderer Sorgfalt, um Vertraulichkeit, Integrität und Verfügbarkeit zu gewährleisten, ohne die Leistungsfähigkeit des Systems zu beeinträchtigen. Mittels eines repräsentativen Anwendungsfalls für realistische Kundenaufträge zum Lasermarkieren wurde das entwickelte System erfolgreich auf seine Anwendbarkeit überprüft. Dabei erfolgte sowohl eine Evaluation der Sicherheit als auch der Leistungsfähigkeit des Assistenzsystems. Die Resultate dieser Überprüfung belegen die Tragfähigkeit der konzipierten und implementierten Lösung.

Aufbauend auf den Ergebnissen dieser Dissertation wurden im Ausblick weiterführende Forschungsfragen sowie Entwicklungspotentiale identifiziert und beschrieben.

9 LITERATUR

[1] H. Hügel and T. Graf, *Laser in der Fertigung*. Wiesbaden: Vieweg+Teubner, 2009. doi: 10.1007/978-3-8348-9570-7.

[2] R. Anderl, "Industrie 4.0 – technological approaches, use cases, and implementation," - *Autom.*, vol. 63, no. 10, pp. 753–765, Oct. 2015, doi: 10.1515/auto-2015-0025.

[3] S. Dreyer, "Digital Transformation in the Manufacturing Industry: Business Models and Smart Service Systems," Universität Hannover, 2020.

[4] A. Schatz and T. Bauernhansl, "Geschäftsmodell-Innovationen," in *Handbuch Industrie 4.0*, B. Vogel-Heuser, T. Bauernhansl, and M. ten Hompel, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2015, pp. 1–15. doi: 10.1007/978-3-662-45537-1_95-1.

[5] R. Anderl *et al.*, *Industrie 4.0 grenzenlos*. in Xpert.press. Berlin: Springer Vieweg, 2016.

[6] A. Dolgin, *Manifesto of the New Economy: Institutions and Business Models of the Digital Society*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012. doi: 10.1007/978-3-642-21277-2.

[7] S. Gatziu Grivas, Ed., *Digital Business Development: Die Auswirkungen der Digitalisierung auf Geschäftsmodelle und Märkte*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2020. doi: 10.1007/978-3-662-59807-8.

[8] T. Wolf and J.-H. Strohschen, "Digitalisierung: Definition und Reife: Quantitative Bewertung der digitalen Reife," *Inform.-Spektrum*, vol. 41, no. 1, pp. 56–64, Feb. 2018, doi: 10.1007/s00287-017-1084-8.

[9] Prof. Dr. T. Eymann and B. Semba, "Auswirkungen der Digitalisierung auf die Datensicherheit," in *Herausforderungen für Familienunternehmen*, K. Windthorst, Ed., Nomos Verlagsgesellschaft mbH & Co. KG, 2020, pp. 21–28. doi: 10.5771/9783845288987-21.

[10] Plattform Industrie 4.0, "Was ist Industrie 4.0?" Accessed: Dec. 27, 2021. [Online]. Available: https://www.plattform-i40.de/IP/Navigation/DE/Industrie40/WasIndustrie40/was-ist-industrie-40.html

[11] Plattform Industrie 4.0, "Manufacturing-X: Initiative zur Digitalisierung der Lieferketten in der Industrie." Bundesministerium für Wirtschaft und Energie (BMWi), 20122. [Online]. Available: https://www.plattform-i40.de/IP/Redaktion/DE/Downloads/Publikation/Leitbild-2030-f%C3%BCr-Industrie-4.0.pdf?__blob=publicationFile&v=11

[12] C. Lanquillon and S. Schacht, "Der Analytics-Marktplatz," in *Blockchain und maschinelles Lernen*, S. Schacht and C. Lanquillon, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2019, pp. 167–193. doi: 10.1007/978-3-662-60408-3_5.

[13] A. Brandão, H. S. Mamede, and R. Gonçalves, “Trusted Data’s Marketplace,” in *New Knowledge in Information Systems and Technologies*, vol. 930, Á. Rocha, H. Adeli, L. P. Reis, and S. Costanzo, Eds., in Advances in Intelligent Systems and Computing, vol. 930. , Cham: Springer International Publishing, 2019, pp. 515–527. doi: 10.1007/978-3-030-16181-1_49.

[14] T. Kollmann, *E-Business: Grundlagen elektronischer Geschäftsprozesse in der Digitalen Wirtschaft*. Wiesbaden: Springer Fachmedien Wiesbaden, 2019. doi: 10.1007/978-3-658-26143-6.

[15] D. Chaffey, *E-business and e-commerce management: strategy, implementation and practice*, 4th ed. Harlow, England ; New York: FT Prentice Hall, 2009.

[16] E. Turban, J. Outland, D. King, J. K. Lee, T.-P. Liang, and D. C. Turban, *Electronic Commerce 2018*. in Springer Texts in Business and Economics. Cham: Springer International Publishing, 2018. doi: 10.1007/978-3-319-58715-8.

[17] B. W. Wirtz, *Digital Business Models: Concepts, Models, and the Alphabet Case Study*, 1st ed. 2019. in Progress in IS. Cham: Springer International Publishing : Imprint: Springer, 2019. doi: 10.1007/978-3-030-13005-3.

[18] C. Stummeyer and B. Köber, Eds., *Amazon für Entscheider: Strategieentwicklung, Implementierung und Fallstudien für Hersteller und Händler*. Wiesbaden: Springer Fachmedien Wiesbaden, 2020. doi: 10.1007/978-3-658-27427-6.

[19] T. Kollmann, *E-Business kompakt: Grundlagen elektronischer Geschäftsprozesse in der Digitalen Wirtschaft mit über 70 Fallbeispielen*. Wiesbaden: Springer Fachmedien Wiesbaden, 2019. doi: 10.1007/978-3-658-26978-4.

[20] G. Ramsdell, “The real business of B2B.” The McKinsey Quarterly 3:174–184, 2000. Accessed: Nov. 05, 2021. [Online]. Available: https://www.proquest.com/openview/f682804f67431debaa12e02d4b053ce9/1?pq-origsite=gscholar&cbl=30375

[22] E. Hartmann, “Ein Überblick der E-Marktplätze im B2B-Bereich,” in *Handbuch Digitale Wirtschaft*, T. Kollmann, Ed., Wiesbaden: Springer Fachmedien Wiesbaden, 2020, pp. 603–629. doi: 10.1007/978-3-658-17291-6_45.

[23] F. Stahl, F. Schomm, G. Vossen, and L. Vomfell, “A classification framework for data marketplaces,” *Vietnam J. Comput. Sci.*, vol. 3, no. 3, pp. 137–143, Aug. 2016, doi: 10.1007/s40595-016-0064-2.

[24] D. Fernandez, A. Futoransky, G. Ajzenman, M. Travizano, and C. Sarraute, “Wibson Protocol for Secure Data Exchange and Batch Payments,” *ArXiv200108832 Cs*, Jan. 2020, Accessed: Aug. 26, 2020. [Online]. Available: http://arxiv.org/abs/2001.08832

[25] M. Spiekermann, “Data Marketplaces: Trends and Monetisation of Data Goods,” *Intereconomics*, vol. 54, no. 4, pp. 208–216, Jul. 2019, doi: 10.1007/s10272-019-0826-z.

[26] A. Ng and K. Soo, *Data Science – was ist das eigentlich?!: Algorithmen des maschinellen Lernens verständlich erklärt*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2018. doi: 10.1007/978-3-662-56776-0.

[27] V. Wittpahl, Ed., *Künstliche Intelligenz: Technologie | Anwendung | Gesellschaft*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2019. doi: 10.1007/978-3-662-58042-4.

[28] P. Buxmann and H. Schmidt, Eds., *Künstliche Intelligenz: Mit Algorithmen zum wirtschaftlichen Erfolg*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2019. doi: 10.1007/978-3-662-57568-0.

[29] J. Lee, *Business hack: the wealth dragon way to build a successful business in the digital age*, First Edition. Hoboken: Wiley, 2019.

[30] S. Mayer, D. Plangger, F. Michahelles, and S. Rothfuss, "UberManufacturing: A Goal-Driven Collaborative Industrial Manufacturing Marketplace," in *Proceedings of the 6th International Conference on the Internet of Things - IoT'16*, Stuttgart, Germany: ACM Press, 2016, pp. 111–119. doi: 10.1145/2991561.2991569.

[31] "Automotive Big Data Marketplace for Innovative Cross-sectorial Vehicle Data Services (AutoMat) Project Presentation." Horizon 2020 European Union Funding for Research & Innovation. Accessed: Sep. 02, 2020. [Online]. Available: https://trimis.ec.europa.eu/sites/default/files/project/documents/AutoMat_ProjectPresentation.pdf

[32] J. Pillmann, C. Wietfeld, A. Zarcula, T. Raugust, and D. C. Alonso, "Novel Common Vehicle Information Model (CVIM) for Future Automotive Vehicle Big Data Marketplaces," *2017 IEEE Intell. Veh. Symp. IV*, pp. 1910–1915, Jun. 2017, doi: 10.1109/IVS.2017.7995984.

[33] J. Bauer, F. Gehrs, M. Jatzlau, and S. Scheuren, "Dezentraler Marktplatz in einer offenen, dezentralen Soft-ware-Plattform für landwirtschaftliche Dienstleistun- gen," p. 4.

[34] P. Sharma, S. Lawrenz, and A. Rausch, "Towards Trustworthy and Independent Data Marketplaces," in *Proceedings of the 2020 The 2nd International Conference on Blockchain Technology*, Hilo HI USA: ACM, Mar. 2020, pp. 39–45. doi: 10.1145/3390566.3391687.

[35] B. Krishnamachari, J. Power, S. H. Kim, and C. Shahabi, "I3: An IoT Marketplace for Smart Communities," in *Proceedings of the 16th Annual International Conference on Mobile Systems, Applications, and Services*, Munich Germany: ACM, Jun. 2018, pp. 498–499. doi: 10.1145/3210240.3223573.

[36] H. J. Eichler and J. Eichler, *Laser*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2015. doi: 10.1007/978-3-642-41438-1.

[37] Prof. Dr.-Ing. Dipl.-Wirtsch.-Ing. P. Groche, "Laser in der Fertigung," Fachgebiet für Produktionstechnik und Umformmaschinen, TU darmstadt, Wintersemester /2019 2018.

[38] H. Hügel and T. Graf, *Laser in der Fertigung: Strahlquellen, Systeme, Fertigungsverfahren*. Wiesbaden: Vieweg + Teubner, 2009. Accessed: Jan. 15, 2021. [Online]. Available: https://doi.org/10.1007/978-3-8348-9570-7

[39] M. G. Müller, *Prozessüberwachung beim Laserstrahlschweißen durch Auswertung der reflektierten Leistung*. in Laser in der Materialbearbeitung. München: Utz, Wiss, 2002.

[40] K. Krastel, *Konzepte und Konstruktionen zur laserintegrierten Komplettbearbeitung in Werkzeugmaschinen*. in Laser in der Materialbearbeitung. München: Utz, 2002.

[41] D. Kochan, Ed., *Solid freeform manufacturing: advanced rapid prototyping*. in Manufacturing research and technology, no. 19. Amsterdam ; New York: Elsevier, 1993.

[42] E. Kannatey-Asibu, *Principles of Laser Materials Processing*. Hoboken, NJ, USA: John Wiley & Sons, Inc., 2009. doi: 10.1002/9780470459300.

[43] T. Graf, *Laser: Grundlagen der Laserstrahlerzeugung*, 2., Überarbeitete und erweiterte Auflage. in Lehrbuch. Wiesbaden: Springer Vieweg, 2015.

[44] J. D. Majumdar and I. Manna, Eds., *Laser-assisted fabrication of materials*. in Springer series in materials science, no. volume 161. New York: Springer, 2013.

[45] R. Menzel, *Photonics*. in Advanced Texts in Physics. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001. doi: 10.1007/978-3-662-04521-3.

[46] B. L. Mordike, A. B. Vannes, and IITT-International, Eds., *Laser-6*. in Technology transfer series. Gournay-sur-Marne, France: IITT-International, 1990.

[47] R. Poprawe, *Lasertechnik für die Fertigung: Grundlagen, Perspektiven und Beispiele für den innovativen Ingenieur ; mit 26 Tabellen*. in VDI-Buch. Berlin: Springer, 2005.

[48] P. Schaaf, Ed., *Laser Processing of Materials*, vol. 139. in Springer Series in Materials Science, vol. 139. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010. doi: 10.1007/978-3-642-13281-0.

[49] M. W. Sigrist, *Laser: Theorie, Typen und Anwendungen*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2018. doi: 10.1007/978-3-662-57515-4.

[50] W. M. Steen, *Laser material processing*, 3rd ed. London ; New York: Springer, 2003.

[51] F. Träger, Ed., *Springer Handbook of Lasers and Optics*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012. doi: 10.1007/978-3-642-19409-2.

[52] M. von Behr and C. (Hrsg) Köhler, "Werkstattoffene CIM-Konzepte. Alternativen für CAD/CAM und Fertigungssteuerung," p. 111, 1990.

[53] S. Vajna, C. Weber, H. Bley, and K. Zeman, *CAx für Ingenieure: eine praxisbezogene Einführung*, 2., Völlig neu bearbeitete Auflage. Berlin Heidelberg: Springer, 2009.

[54] T. Bauernhansl, "Die Vierte Industrielle Revolution – Der Weg in ein wertschaffendes Produktionsparadigma," in *Industrie 4.0 in Produktion, Automatisierung und Logistik*, T. Bauernhansl, M. ten Hompel, and B. Vogel-Heuser, Eds., Wiesbaden: Springer Fachmedien Wiesbaden, 2014, pp. 5–35.

[55] P. Hehenberger, *Computerunterstützte Fertigung*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011. doi: 10.1007/978-3-642-13475-3.

[56] M. Weck, C. Brecher, and M. Weck, *Automatisierung von Maschinen und Anlagen*, 6., neu Bearb. Aufl. in Werkzeugmaschinen, no. Manfred Weck; Christian Brecher ; 4. Berlin: Springer, 2006.

[57] A. Giehl *et al.*, "Analysen, Anforderungen und Konzeptentwicklung für den Technologiedatenmarktplatz." 2016.

[58] "IUNO-Glossar." Jan. 03, 2018. Accessed: Jul. 06, 2022. [Online]. Available: https://iuno-projekt.de/glossar

[59] "Das Projekt: Nationalen Referenzprojekts zur IT-Sicherheit in Industrie 4.0 – IUNO." Accessed: May 02, 2023. [Online]. Available: https://iuno-projekt.de/das-projekt

[60] A. Giehl *et al.*, "Stand der Technik, Bedrohungs- und Risikoanalyse zum Demonstrator „Technologiedatenmarktplatz"." IUNO – Nationales Referenzprojekt IT-Sicherheit in Industrie 4.0, 2016.

[61] accessec GmbH *et al.*, "Forschung für mehr IT-Sicherheit in Industrie 4.0 (IUNO Projektergebnisse)." 2018.

[62] G. Shaabany, M. Grimm, and R. Anderl, "Secure Information Model for Data Marketplaces enabling Global Distributed Manufacturing," *Proc. 26th CIRP Des. Conf.*, Jun. 2016, [Online]. Available: http://tubiblio.ulb.tu-darmstadt.de/81252/

[63] G. Shaabany, S. Frisch, and R. Anderl, "Secure Concept for Online Trading of Technology Data in Global Manufacturing Market," in *Product Lifecycle Management and the Industry of the Future*, vol. 517, J. Ríos, A. Bernard, A. Bouras, and S. Foufou, Eds., Cham: Springer International Publishing, 2017, pp. 690–700. doi: 10.1007/978-3-319-72905-3_61.

[64] G. Shaabany and R. Anderl, "Reliable Innovative Business Model for Online Trading of Machines' Parameters in the Automation and Manufacturing Sector," p. 8, 2017.

[65] Wibu-Systems, "CmStick | CmStick ME." Wibu-Systems. [Online]. Available: https://cdn.wibu.com/fileadmin/wibu_downloads/CodeMeter_Datasheets/Seriennummer_03/ME/CmStick_ME-1001-03-1x0_EN.pdf

[66] B. G. Buchanan and E. H. Shortliffe, Eds., *Rule-based expert systems: the MYCIN experiments of the Stanford Heuristic Programming Project*. in The Addison-Wesley series in artificial intelligence. Reading, Mass: Addison-Wesley, 1984.

[67] E. A. Feigenbaum, P. McCorduck, and H. P. Nii, *The rise of the expert company: how visionary companies are using artificial intelligence to achieve higher productivity and profits*. London: Macmillan London, 1988.

[68] Platform Industrie 4.0, *Sichere Kommunikation für Industrie 4.0: Diskussionspapier*. Bundesministerium für Wirtschaft und Energie (BMWi), 2017. Accessed: Dec. 15, 2020. [Online]. Available: https://www.plattform-i40.de/PI40/Redaktion/DE/Downloads/Publikation/sichere-kommunikation-i40.pdf?__blob=publicationFile&v=5

[69] R. Ross, M. McEvilley, and J. C. Oren, "Systems security engineering: considerations for a multidisciplinary approach in the engineering of trustworthy secure systems,

volume 1," National Institute of Standards and Technology, Gaithersburg, MD, NIST SP 800-160v1, 2016. doi: 10.6028/NIST.SP.800-160v1.

[70] "BSI - Glossar der Cyber-Sicherheit - I." Accessed: Nov. 07, 2020. [Online]. Available: https://www.bsi.bund.de/DE/Themen/Cyber-Sicherheit/Empfehlungen/cyberglossar/Functions/glossar.html?cms_lv2=9817288

[71] C. Eckert, "IT-Sicherheit, Konzepte – Verfahren – Protokolle," München, 2013.

[72] M. Bedner and T. Ackermann, "Schutzziele der IT-Sicherheit," *Datenschutz Datensicherheit - DuD*, vol. 34, no. 5, pp. 323–328, May 2010, doi: 10.1007/s11623-010-0096-1.

[73] DIN Deutsches Institut für Normung, *Informationstechnik – IT-Sicherheitsverfahren – Informationssicherheits-Managementsysteme – Überblick und Terminologie*. Berlin: Normenausschuss Informationstechnik und Anwendungen (NIA) im DIN, 2009.

[74] C. Eckert, *IT-Sicherheit: Konzepte, Verfahren, Protokolle*, 10. Auflage. in De Gruyter studium. München: De Gruyter Oldenburg, 2018.

[75] S. Spitz, M. Pramateftakis, and J. Swoboda, *Kryptographie und IT-Sicherheit*. Wiesbaden: Vieweg+Teubner, 2011. doi: 10.1007/978-3-8348-8120-5.

[76] K. Böttinger, B. Filipovic, M. Hutle, S. Rohr, Fraunhofer-Einrichtung für Angewandte und Integrierte Sicherheit AISEC, and accessec GmbH, *Leitfaden Industrie 4.0 Security: Handlungsempfehlungen für den Mittelstand*. 2016.

[77] J. Buchmann, "Einführung in die Kryptographie," 6., Überarbeitete Auflage., Berlin; Heidelberg: Springer Spektrum, 2016.

[78] D. Wätjen, *Kryptographie*. Wiesbaden: Springer Fachmedien Wiesbaden, 2018. doi: 10.1007/978-3-658-22474-5.

[79] S. Rubinstein-Salzedo, *Cryptography*, 1st ed. 2018. in Springer Undergraduate Mathematics Series. Cham: Springer International Publishing : Imprint: Springer, 2018. doi: 10.1007/978-3-319-94818-8.

[80] K.-R. Müller, *IT-Sicherheit mit System: integratives IT-Sicherheits-, Kontinuitäts- und Risikomanagement - Sichere Anwendungen - Standards und Practices*, 6., Erweiterte und überarbeitete Auflage. Wiesbaden [Heidelberg]: Springer Vieweg, 2018. doi: 10.1007/978-3-658-22065-5.

[81] W. Ertel, *Angewandte Kryptographie*, 2., Bearb. Aufl. München Wien: Hanser, 2003.

[82] National Institute of Standards and Technology, "Federal Information Processing Standards Publication 197-Announcing the Advanced Encryption Standard (AES)." 2001.

[83] A. Beutelspacher, *Kryptologie*. Wiesbaden: Springer Fachmedien Wiesbaden, 2015. doi: 10.1007/978-3-658-05976-7.

[84] A. J. Menezes, P. C. Van Oorschot, and S. A. Vanstone, *Handbook of applied cryptography*. in CRC Press series on discrete mathematics and its applications. Boca Raton: CRC Press, 1997.

[85] A. K. Lenstra and E. R. Verheul, "Selecting Cryptographic Key Sizes," *J. Cryptol.*, vol. 14, pp. 255–293, 2001.

[86] International Organization for Standardization, "ISO/IEC 9594-8: Information technology — Open systems interconnection — Part 8: The Directory: Public-key and attribute certificate frameworks." 2020. Accessed: Aug. 30, 2022. [Online]. Available: https://webstore.iec.ch/p-preview/info_isoiec9594-8%7Bed8.0%7Den.pdf

[87] A. Ismailisufi, T. Popović, N. Gligorić, S. Radonjic, and S. Šandi, "A Private Blockchain Implementation Using Multichain Open Source Platform," in *2020 24th International Conference on Information Technology (IT)*, Feb. 2020, pp. 1–4. doi: 10.1109/IT48810.2020.9070689.

[88] K. Werbach, *The Blockchain and the New Architecture of Trust*. MIT Press, 2018.

[89] M. Pilkington, "Blockchain Technology: Principles and Applications." Rochester, NY, Sep. 18, 2015. Accessed: Apr. 18, 2023. [Online]. Available: https://papers.ssrn.com/abstract=2662660

[90] V. Schlatt, A. Schweizer, Prof. Dr. N. Urbach, and Prof. Dr. G. Fridgen, "BLOCKCHAIN: GRUNDLAGEN,ANWENDUNGEN UND POTENZIALE." Fraunhofer-Institut für Angewandte Informationstechnik FIT, 2016. [Online]. Available: https://www.fit.fraunhofer.de/content/dam/fit/de/documents/Blockchain_WhitePaper_Grundlagen-Anwendungen-Potentiale.pdf

[91] S. Demirkan, I. Demirkan, and A. McKee, "Blockchain technology in the future of business cyber security and accounting," *J. Manag. Anal.*, vol. 7, no. 2, pp. 189–208, Apr. 2020, doi: 10.1080/23270012.2020.1731721.

[92] M. Yassine, M. Shojafar, M. Alazab, I. Romdhani, and U. KC, *Blockchain for Cybersecurity and Privacy: Architectures, Challenges, and Applications*. 2020.

[93] A. Narayanan, *Bitcoin and cryptocurrency technologies: a comprehensive introduction*. Princeton: Princeton University Press, 2016.

[94] N. Trojanowska, M. Kedziora, M. Hanif, and H. Song, "Secure Decentralized Application Development of Blockchain-based Games," in *2020 IEEE 39th International Performance Computing and Communications Conference (IPCCC)*, Nov. 2020, pp. 1–8. doi: 10.1109/IPCCC50635.2020.9391556.

[95] R. Prakash, V. S. Anoop, and S. Asharaf, "Blockchain technology for cybersecurity: A text mining literature analysis," *Int. J. Inf. Manag. Data Insights*, vol. 2, no. 2, p. 100112, Nov. 2022, doi: 10.1016/j.jjimei.2022.100112.

[96] S. Tern, "Survey of Smart Contract Technology and Application Based on Blockchain," *Open J. Appl. Sci.*, vol. 11, no. 10, pp. 1135–1148, 2021, doi: 10.4236/ojapps.2021.1110085.

[97] T. Hewa, M. Ylianttila, and M. Liyanage, “Survey on blockchain based smart contracts: Applications, opportunities and challenges,” *J. Netw. Comput. Appl.*, vol. 177, p. 102857, Mar. 2021, doi: 10.1016/j.jnca.2020.102857.

[98] Z. Zheng *et al.*, “An Overview on Smart Contracts: Challenges, Advances and Platforms,” *Future Gener. Comput. Syst.*, vol. 105, pp. 475–491, Apr. 2020, doi: 10.1016/j.future.2019.12.019.

[99] W. Zou *et al.*, “Smart Contract Development: Challenges and Opportunities,” *IEEE Trans. Softw. Eng.*, vol. 47, no. 10, pp. 2084–2106, Oct. 2021, doi: 10.1109/TSE.2019.2942301.

[100] J. Froese and S. Straub, “Wem gehören die Daten? Vertragliche Regelungen, Möglichkeiten und Grenzen bei der Nutzung datenbasierter Produkte,” in *Digitalisierung souverän gestalten II*, E. A. Hartmann, Ed., Berlin, Heidelberg: Springer Berlin Heidelberg, 2022, pp. 136–151. doi: 10.1007/978-3-662-64408-9_11.

[101] T. Sassenberg and T. Faber, Eds., *Rechtshandbuch Industrie 4.0 und Internet of Things: Praxisfragen und Perspektiven der digitalen Zukunft*, 2. Auflage. München : [München]: C.H. Beck ; Vahlen, 2020.

[102] S. Ventroni, “Copyrights und Lizenzmanagement,” in *Ökonomie der Musikindustrie*, M. Clement, O. Schusser, and D. Papies, Eds., Wiesbaden: Gabler, 2008, pp. 59–76. doi: 10.1007/978-3-8349-9916-0_5.

[103] T. Groll, *1x1 des Lizenzmanagements: Praxisleitfaden für Lizenzmanager ; [im Internet: hilfreiche Checklisten, Formulare und ergänzende Informationen]*, 2., erw. Aufl. München: Hanser, 2012.

[104] C. Demant, *Erfolgreich ein Software-Startup gründen: Tipps und Erfahrungen eines Tech-Unternehmers*, 2., Erweiterte Auflage. Berlin [Heidelberg]: Springer Gabler, 2020.

[105] D. Gull, “Bewertung von Discountoptionen bei Softwarelizenzverträgen,” *WIRTSCHAFTSINFORMATIK*, vol. 53, no. 4, pp. 213–223, Aug. 2011, doi: 10.1007/s11576-011-0283-1.

[106] S. Brassel and A. Gadatsch, *Softwarelizenzmanagement kompakt: Einsatz und Management des immateriellen Wirtschaftsgutes Software und hybrider Leistungsbündel (Public Cloud Services)*. in IT kompakt. Wiesbaden [Heidelberg]: Springer Vieweg, 2019.

[107] European IP Helpdesk, “Commercialising Intellectual Property: Licence Agreements.” 2021.

[108] R. M. Bürkner, *Erfolgreiche Software-Lizenzierung: Electronic License Management - Von der Auswahl bis zur Installation*. Springer-Verlag, 2013.

[109] “Duden | Lizenz | Rechtschreibung, Bedeutung, Definition, Herkunft.” Accessed: Sep. 01, 2022. [Online]. Available: https://www.duden.de/rechtschreibung/Lizenz

[110] C. Fan and Z. Zhang, “International licensing and R&D subsidy,” p. 34.

[111] D. Gull and A. Wehrmann, “Optimierte Softwarelizenzierung – Kombinierte Lizenztypen im Lizenzportfolio,” *WIRTSCHAFTSINFORMATIK*, vol. 51, no. 4, pp. 324–334, Aug. 2009, doi: 10.1007/s11576-009-0182-x.

[112] W.-S. AG, “CodeMeter Lizenzmodelle.” Accessed: Aug. 19, 2021. [Online]. Available: https://www.wibu.com/de/produkte/codemeter/lizenzmodelle.html

[113] S. Brassel and A. Gadatsch, “Softwarenutzung im Umbruch: Von der Software-Lizenz zum Cloudbasierten Business Process Outsourcing,” *HMD Prax. Wirtsch.*, vol. 54, no. 1, pp. 156–164, Feb. 2017, doi: 10.1365/s40702-016-0279-9.

[114] B. Bodó, D. Gervais, and J. P. Quintais, “Blockchain and smart contracts: the missing link in copyright licensing?,” *Int. J. Law Inf. Technol.*, vol. 26, no. 4, pp. 311–336, Dec. 2018, doi: 10.1093/ijlit/eay014.

[115] A. Kumar, A. Gupta, L. M. Sanagavarapu, and Y. R. Reddy, “An approach to Open-Source Software License Management using Blockchain-based Smart-Contracts,” in *15th Innovations in Software Engineering Conference*, Gandhinagar India: ACM, Feb. 2022, pp. 1–5. doi: 10.1145/3511430.3511448.

[116] V. Urovi, V. Jaiman, A. Angerer, and M. Dumontier, “LUCE: A blockchain-based data sharing platform for monitoring data License accoUntability and CompliancE,” *Blockchain Res. Appl.*, vol. 3, no. 4, p. 100102, Dec. 2022, doi: 10.1016/j.bcra.2022.100102.

[117] F. K. Nishi *et al.*, “Electronic Healthcare Data Record Security Using Blockchain and Smart Contract,” *J. Sens.*, vol. 2022, pp. 1–22, May 2022, doi: 10.1155/2022/7299185.

[118] D. Gatta, K. Hinteregger, and A. Fensel, “Making Licensing of Content and Data Explicit with Semantics and Blockchain,” in *Information Management and Big Data*, vol. 1577, J. A. Lossio-Ventura, J. Valverde-Rebaza, E. Díaz, D. Muñante, C. Gavidia-Calderon, A. D. B. Valejo, and H. Alatrista-Salas, Eds., in Communications in Computer and Information Science, vol. 1577. , Cham: Springer International Publishing, 2022, pp. 370–379. doi: 10.1007/978-3-031-04447-2_25.

[119] M. Schäffner, C. Lichti, J. Gross, and P. Sandner, “KOSMoS Private Blockchain Toolkit: How to Use Hyperledger in an Industrial DLT Project,” 2021.

[120] J. L. Staud, *Geschäftsprozessanalyse: ereignisgesteuerte Prozessketten und objektorientierte Geschäftsprozessmodellierung für betriebswirtschaftliche Standardsoftware*, 3. Aufl. Berlin: Springer, 2006.

[121] Prof. Dr.-Ing. R. Anderl, *Virtuelle Produktentwicklung C-Produkt- und Prozessmodellierung*. 2020.

[122] A. Watson, “Visual Modelling: past, present and future,” p. 6.

[123] R. Braun, W. Esswein, and S. Greiffenberg, *Einführung in die Programmierung: Grundlagen, Java, UML*. in Springer-Lehrbuch. Berlin: Springer, 2006.

[124] C. Rupp and S. Queins, *UML 2 glasklar: Praxiswissen für die UML-Modellierung*, 4., Aktualisierte und erweiterte Auflage. München: Hanser, 2012.

[125] G. Booch and G. Booch, Eds., *Object-oriented analysis and design with applications*, 3rd ed. in The Addison-Wesley object technology series. Upper Saddle River, NJ: Addison-Wesley, 2007.

[126] A. Schwegmann, "Objektorientierte Referenzmodellierung," in *Objektorientierte Referenzmodellierung*, Wiesbaden: Deutscher Universitätsverlag, 1999, pp. 105–184. doi: 10.1007/978-3-322-99774-6_6.

[127] G. Booch, J. Rumbaugh, and I. Jacobson, *Das UML Benutzerhandbuch: aktuell zu Version 2.0*. in Programmer's Choice Addison Wesley. München: Addison Wesley in Pearson Education Deutschland, 2006.

[128] W. Czuchra, *UML in logistischen Prozessen: graphische Sprache zur Modellierung der Systeme ; mit 4 Tabellen ; [mit Online-Service]*, 1. Aufl. in Studium. Wiesbaden: Vieweg + Teubner, 2010.

[129] S. Kleuker, *Grundkurs Software-Engineering mit UML*. Wiesbaden: Springer Fachmedien Wiesbaden, 2018. doi: 10.1007/978-3-658-19969-2.

[130] J. Seemann and J. W. von Gudenberg, *Software-Entwurf mit UML 2: objektorientierte Modellierung mit Beispielen in Java*. Berlin: Springer, 2006. Accessed: Oct. 20, 2020. [Online]. Available: http://public.eblib.com/choice/publicfullrecord.aspx?p=417841

[131] M. Seidl, M. Scholz, C. Huemer, and G. Kappel, *UML @ Classroom*. in Undergraduate Topics in Computer Science. Cham: Springer International Publishing, 2015. doi: 10.1007/978-3-319-12742-2.

[132] T. Weilkiens, *Systems engineering mit SysML/UML: Modellierung, Analyse, Design*, 1. Aufl. Heidelberg: dpunkt.verl, 2006.

[133] B. Oestereich and A. Scheithauer, *Analyse und Design mit der UML 2.5: objektorientierte Softwareentwicklung*, 11., Umfassend überarb. und aktualisierte Aufl. München: Oldenbourg, 2013.

[134] C. Rupp and SOPHIST-Gesellschaft für Innovatives Software-Engineering, Eds., *Requirements-Engineering und -Management: professionelle, iterative Anforderungsanalyse für die Praxis*, 4., Aktualisierte und erw. Aufl. München: Hanser, 2007.

[135] C. Ebert, *Systematisches Requirements Engineering und Management: Anforderungen ermitteln, spezifizieren, analysieren und verwalten*, 2., Aktualisierte und erw. Aufl. Heidelberg: dpunkt-Verl, 2008.

[136] J. M. Fernandes and R. J. Machado, *Requirements in Engineering Projects*. in Lecture Notes in Management and Industrial Engineering. Cham: Springer International Publishing, 2016. doi: 10.1007/978-3-319-18597-2.

[137] J. Holt, S. A. Perry, and M. Brownsword, *Model-Based Requirements Engineering*. Institution of Engineering and Technology, 2011. doi: 10.1049/PBPC009E.

[138] D. Kulak and E. Guiney, *Use cases: requirements in context*. Reading, Mass: Addison-Wesley, 2000.

[139] M. Gottschalk, C. Delfs, and M. Uslar, *The Use Case and Smart Grid Architecture Model Approach: The IEC 62559-2 Use Case Template and the SGAM applied in various domains*, 1st ed. 2017. in SpringerBriefs in Energy. Cham: Springer International Publishing : Imprint: Springer, 2017. doi: 10.1007/978-3-319-49229-2.

[140] ISO/IEC, *ISO/IEC 19505-2: 2012 Information Technology – Object Management Group Unified Modeling Language (OMG UML) – Part 2: Superstructure*. 2012.

[141] K. Friedrich Gebhardt, "UML Unified Modeling Language." Accessed: Oct. 21, 2020. [Online]. Available: http://wwwlehre.dhbw-stuttgart.de/~kfg/uml/uml.pdf

[142] A. van Lamsweerde, "Elaborating security requirements by construction of intentional anti-models," in *Proceedings. 26th International Conference on Software Engineering*, May 2004, pp. 148–157. doi: 10.1109/ICSE.2004.1317437.

[143] R. Rieke, "Security Analysis of System Behaviour," Dissertation.

[144] BSI, *Gefährdungskatalog-G0-Elementare Gefährdungen*, vol. 12. EL. in IT-Grundschutz-Kataloge, vol. 12. EL. 2011. Accessed: Dec. 03, 2020. [Online]. Available: https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Grundschutz/Download/Gefaehrdungskatalog-G0-ElementareGefaehrdungen.pdf;jsessionid=2A6862F3CC22E43EA502556A5FF09093.1_cid502?__blob=publicationFile&v=1

[145] S. Dukanovic, M. Matthess, R. Schwarz, S. Frisch, and A. Borisov, *Bedrohungsmodell fü Wertschöpfungsnetzwerke der Industrie 4.0*. in D6.1-1 / M 06. IUNO – Nationales Referenzprojekt IT-Sicherheit in Industrie 4.0.

[146] A. Shostack, *Threat modeling: designing for security*. Indianapolis, IN: Wiley, 2014.

[147] Z. Ma and C. Schmittner, *Threat Modeling for Automotive Security Analysis*. 2016, p. 339. doi: 10.14257/astl.2016.139.68.

[148] Bundesanzeiger Verlag GmbH, Deutschland, and Bundesamt für Sicherheit in der Informationstechnik, *IT-Grundschutz-Kompendium*. 2020.

[149] G. Pahl, W. Beitz, J. Feldhusen, and K.-H. Grote, Eds., *Konstruktionslehre: Grundlagen erfolgreicher Produktentwicklung ; Methoden und Anwendung*, 7. Aufl. in Springer-Lehrbuch. Berlin: Springer, 2007.

[150] K. Grąbczewski, *Meta-Learning in Decision Tree Induction*, vol. 498. in Studies in Computational Intelligence, vol. 498. Cham: Springer International Publishing, 2014. doi: 10.1007/978-3-319-00960-5.

[151] C. Beierle and G. Kern-Isberner, *Methoden wissensbasierter Systeme: Grundlagen - Algorithmen - Anwendungen*, 3., erw. Aufl. in Computational intelligence. Wiesbaden: Vieweg, 2006.

[152] H. Dahan, S. Cohen, L. Rokach, and O. Maimon, *Proactive Data Mining with Decision Trees*. in SpringerBriefs in Electrical and Computer Engineering. New York, NY: Springer New York, 2014. doi: 10.1007/978-1-4939-0539-3.

[153] M. Bramer, *Principles of Data Mining*. in Undergraduate Topics in Computer Science. London: Springer London, 2020. doi: 10.1007/978-1-4471-7493-6.

[154] J. Daor, J. Daemen, and V. Rijmen, "AES proposal: rijndael," Oct. 1999.

[155] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Commun. ACM*, vol. 21, no. 2, Art. no. 2, Feb. 1978, doi: 10.1145/359340.359342.

[156] D. R. S. Paterson Maura, *Cryptography: Theory and Practice*, 4th ed. New York: Chapman and Hall/CRC, 2018. doi: 10.1201/9781315282497.

[157] E. B. Barker and Q. H. Dang, "Recommendation for Key Management Part 3: Application-Specific Key Management Guidance," National Institute of Standards and Technology, NIST SP 800-57Pt3r1, Jan. 2015. doi: 10.6028/NIST.SP.800-57Pt3r1.

[158] "Cryptography Engineering: Design Principles and Practical Applications | Wiley," Wiley.com. Accessed: Jun. 27, 2023. [Online]. Available: https://www.wiley.com/en-be/Cryptography+Engineering%3A+Design+Principles+and+Practical+Applications+-p-9780470474242

[159] R. L. Ashenhurst, "Ontological aspects of information modeling," *Minds Mach.*, vol. 6, no. 3, pp. 287–394, Oct. 1996, doi: 10.1007/BF00729802.

[160] ISO/IEC, "ISO/IEC 11889: Information technology — Trusted Platform Module — Part 1: Overview." 2009. Accessed: Jun. 14, 2023. [Online]. Available: https://www.iso.org/standard/50970.html

[161] "What is Web Application (Web Apps) and its Benefits," Software Quality. Accessed: Jun. 22, 2023. [Online]. Available: https://www.techtarget.com/searchsoftwarequality/definition/Web-application-Web-app

[162] "How to Make a Desktop Application? A Detailed Guide," Radixweb. Accessed: Jun. 22, 2023. [Online]. Available: https://radixweb.com/blog/desktop-application-development-guide

[163] M. Shaw and D. Garlan, "Software Architecture: Perspectives on an Engineering Discipline," p. 0 Bytes, 1996, doi: 10.1184/R1/6625796.V1.

[164] IEEE Computer Society, "ISO/IEC 15288:2008(E): Systems and software engineering - System life cycle processes," *Softw. Syst. Eng. Stand. Comm.*, vol. ISO/IEC 15288:2008, no. IEEE Std 15288:2008, 2008.

[165] ISO/IEC/IEEE 42010, *Systems and software engineering — Architecture description*. in ISO/IEC/IEEE 42010. 2011.

[166] Atlassian, "Microservices vs. monolithic architecture," Atlassian. Accessed: Jul. 13, 2023. [Online]. Available: https://www.atlassian.com/microservices/microservices-architecture/microservices-vs-monolith

[167] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System".

[168] V. Buterin, “Ethereum: A Next-Generation Smart Contract and Decentralized Application Platform.,” 2015.

[169] Toshendra Sharma, “Types of Blockchains Explained- Public Vs. Private Vs. Consortium.” Accessed: Apr. 20, 2023. [Online]. Available: https://www.blockchain-council.org/blockchain/types-of-blockchains-explained-public-vs-private-vs-consortium/

[170] E. Androulaki, S. Cocco, C. F. P. May 10, and 2018, “IBM Developer,” IBM Developer. Accessed: Apr. 20, 2023. [Online]. Available: https://developer.ibm.com/tutorials/cl-blockchain-private-confidential-transactions-hyperledger-fabric-zero-knowledge-proof/

[171] “Channels — hyperledger-fabricdocs main documentation.” Accessed: Apr. 20, 2023. [Online]. Available: https://hyperledger-fabric.readthedocs.io/en/latest/channels.html

[172] “Flows,” R3 Documentation. Accessed: Apr. 20, 2023. [Online]. Available: https://docs.r3.com/en/platform/corda/4.9/enterprise/cordapps/api-flows.html

[173] “Peers — hyperledger-fabricdocs main documentation.” Accessed: Jun. 05, 2023. [Online]. Available: https://hyperledger-fabric.readthedocs.io/en/latest/peers/peers.html

[174] “Fabric Gateway — hyperledger-fabricdocs main documentation.” Accessed: May 23, 2023. [Online]. Available: https://hyperledger-fabric.readthedocs.io/en/latest/gateway.html?highlight=Fabric%20Gateway

[175] “Identity — hyperledger-fabricdocs main documentation.” Accessed: May 23, 2023. [Online]. Available: https://hyperledger-fabric.readthedocs.io/en/latest/identity/identity.html

[176] “Membership Service Provider (MSP) — hyperledger-fabricdocs main documentation.” Accessed: May 23, 2023. [Online]. Available: https://hyperledger-fabric.readthedocs.io/en/latest/membership/membership.html

[177] “The Ordering Service — hyperledger-fabricdocs main documentation.” Accessed: May 22, 2023. [Online]. Available: https://hyperledger-fabric.readthedocs.io/en/latest/orderer/ordering_service.html

[178] “How Fabric networks are structured — hyperledger-fabricdocs main documentation.” Accessed: May 23, 2023. [Online]. Available: https://hyperledger-fabric.readthedocs.io/en/latest/network/network.html

[179] D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley, and W. Polk, “Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile,” RFC Editor, RFC5280, May 2008. doi: 10.17487/rfc5280.

[180] “Policies — hyperledger-fabricdocs main documentation.” Accessed: Jun. 06, 2023. [Online]. Available: https://hyperledger-fabric.readthedocs.io/en/release-2.5/policies/policies.html#how-are-policies-implemented

[181] “Getting Started - Install — hyperledger-fabricdocs main documentation.” Accessed: May 23, 2023. [Online]. Available: https://hyperledger-fabric.readthedocs.io/en/release-2.5/getting_started.html

[182] “Commands Reference — hyperledger-fabricdocs main documentation.” Accessed: May 23, 2023. [Online]. Available: https://hyperledger-fabric.readthedocs.io/en/release-2.5/command_ref.html

[183] “gRPC,” gRPC. Accessed: Jun. 09, 2023. [Online]. Available: https://grpc.io/

[184] R. P. Ltd, “Operating system images,” Raspberry Pi. Accessed: Aug. 22, 2023. [Online]. Available: https://www.raspberrypi.com/software/operating-systems/#raspberry-pi-os-64-bit

[185] I. T. AG, “OPTIGA™ TPM SLM 9670 - Infineon Technologies.” Accessed: Jun. 14, 2023. [Online]. Available: https://www.infineon.com/cms/en/product/security-smart-card-solutions/optiga-embedded-security-solutions/optiga-tpm/slm-9670/

[186] “Python Knowledge Engine (PyKE),” SourceForge. Accessed: Jun. 09, 2023. [Online]. Available: https://sourceforge.net/projects/pyke/

[187] “Welcome to Flask — Flask Documentation (1.1.x).” Accessed: Feb. 16, 2021. [Online]. Available: https://flask.palletsprojects.com/en/1.1.x/

[188] “Welcome to Pyramid, a Python Web Framework.” Accessed: Jun. 26, 2023. [Online]. Available: https://trypyramid.com/

[189] “Django,” Django Project. Accessed: Jun. 26, 2023. [Online]. Available: https://www.djangoproject.com/

[190] M. Solutions, “Pyramid vs Django vs Flask | Flask vs Django vs Pyramid | Mindfire Solutions,” Blogs @ Mindfire Solutions. Accessed: Jun. 26, 2023. [Online]. Available: https://www.mindfiresolutions.com/blog/2018/01/pyramid-vs-django/

[191] “PostgreSQL 13.1 Documentation,” PostgreSQL Documentation. Accessed: Feb. 10, 2021. [Online]. Available: https://www.postgresql.org/docs/13/index.html

[192] “Home - Django REST framework.” Accessed: Jun. 26, 2023. [Online]. Available: https://www.django-rest-framework.org/

[193] “JavaScript With Syntax For Types.” Accessed: Jun. 27, 2023. [Online]. Available: https://www.typescriptlang.org/

[194] “React.” Accessed: Jun. 27, 2023. [Online]. Available: https://react.dev/

[195] “React Admin.” Accessed: Jun. 27, 2023. [Online]. Available: https://marmelab.com/react-admin

[196] “MUI: The React component library you always wanted.” Accessed: Jun. 27, 2023. [Online]. Available: https://mui.com/

[197] “DIN EN ISO 9000:2015-11, Qualitätsmanagementsysteme_- Grundlagen und Begriffe (ISO_9000:2015); Deutsche und Englische Fassung EN_ISO_9000:2015,” Beuth Verlag GmbH. doi: 10.31030/2325650.

[198] S. Brugger-Gebhardt, *Die DIN EN ISO 9001:2015 verstehen*. Wiesbaden: Springer Fachmedien Wiesbaden, 2016. doi: 10.1007/978-3-658-14495-1.

[199] B. Böttcher and M. Nüttgens, “Überprüfung der Gebrauchstauglichkeit von Anwendungssoftware,” *HMD Prax. Wirtsch.*, vol. 50, no. 6, pp. 16–25, Oct. 2013, doi: 10.1007/BF03342065.

[200] “TruMark Station 3000 von TRUMPF.” Accessed: Aug. 22, 2023. [Online]. Available: https://www.trumpf.com/de_DE/produkte/maschinen-systeme/2d-laserschneidmaschinen/trulaser-center-7030/

[201] R. P. Ltd, “Raspberry Pi 4 Model B specifications,” Raspberry Pi. Accessed: Aug. 24, 2023. [Online]. Available: https://www.raspberrypi.com/products/raspberry-pi-4-model-b/specifications/

[202] “Amazon EC2 T2 Instances – Amazon Web Services (AWS),” Amazon Web Services, Inc. Accessed: Aug. 24, 2023. [Online]. Available: https://aws.amazon.com/ec2/instance-types/t2/

[203] X. Lang and W. Mao, “Big Data-Based Decision Support Systems,” in *Encyclopedia of Ocean Engineering*, W. Cui, S. Fu, and Z. Hu, Eds., Singapore: Springer Nature Singapore, 2022, pp. 143–149. doi: 10.1007/978-981-10-6946-8_255.

[204] W. Cui, S. Fu, and Z. Hu, Eds., “Machine Learning,” in *Encyclopedia of Ocean Engineering*, Singapore: Springer Nature Singapore, 2022, pp. 957–957. doi: 10.1007/978-981-10-6946-8_300435.

[205] P. Fischer, *Algorithmisches Lernen*. in Leitfäden der Informatik. Wiesbaden: Vieweg+Teubner Verlag, 1999. doi: 10.1007/978-3-663-11956-2.

10 ANHANG

A. 1: Detaillierte Beschreibung der Einrichtung des „Hyperledger Fabric“-Netzwerks für die Implementierung des Lizenz-Vertrauensagenten.

1. Erstellen eines neuen Verzeichnisses im Home-Ordner des Anwenders

```
mkdir ~/license-trusted-agent && cd ~/license-trusted-agent
```

2. Installieren der benötigten Pakete

```
sudo apt-get install git curl
install docker and docker-compose
sudo systemctl start docker
sudo systemctl enable docker
curl -sSLO https://raw.githubusercontent.com/hyperledger/fabric/main/scripts/install-fabric.sh &&
chmod +x install-fabric.sh
./install-fabric.sh docker binary
```

3. Erstellen der Datei *crypto-config.yaml* für jede Organisation und jeden Auftraggeber im Verzeichnis *config/cryptogen*

- Erstellen der erforderlichen Dateien und Verzeichnisse

```
mkdir -p config/cryptogen
touch config/cryptogen/crypto-config-manufacturer-a.yaml
touch config/cryptogen/crypto-config-user-x.yaml
touch config/cryptogen/crypto-config-orderer.yaml
```

- Kopieren der folgenden Konfiguration und Einfügen in die Datei *crypto-config-manufacturer-a.yaml*

```
nano config/cryptogen/crypto-config-manufacturer-a.yaml
# ---------------------------------------------------------------------------
# "PeerOrgs" - Definition of organizations managing peer nodes
# ---------------------------------------------------------------------------
PeerOrgs:
  # ---------------------------------------------------------------------------
  # ManufacturerA
  # ---------------------------------------------------------------------------
  - Name: ManufacturerA
    Domain: manufacturer-a.laser-lta.com
    EnableNodeOUs: true
    Template:
      Count: 1
      SANS:
        - localhost
    # ---------------------------------------------------------------------------
    # "Users"
    # ---------------------------------------------------------------------------
    # Count: The number of user accounts _in addition_ to Admin
    # ---------------------------------------------------------------------------
    Users:
      Count: 1
```

- Kopieren der folgenden Konfiguration und Einfügen in die Datei *crypto-config-user-x.yaml*

```
nano config/cryptogen/crypto-config-user-x.yaml
# ---------------------------------------------------------------------------
# "PeerOrgs" - Definition of organizations managing peer nodes
# ---------------------------------------------------------------------------
PeerOrgs:
  # ---------------------------------------------------------------------------
  # ManufacturerB
  # ---------------------------------------------------------------------------
  - Name: UserX
    Domain: user-x.laser-lta.com
    EnableNodeOUs: true
    Template:
      Count: 1
      SANS:
        - localhost
    # ---------------------------------------------------------------------------
    # "Users"
    # ---------------------------------------------------------------------------
    # Count: The number of user accounts _in addition_ to Admin
    # ---------------------------------------------------------------------------
    Users:
      Count: 1
```

- Kopieren der folgenden Konfiguration und Einfügen in die Datei *crypto-config-orderer.yaml*

```
nano config/cryptogen/crypto-config-orderer.yaml
# ---------------------------------------------------------------------------
# "OrdererOrgs" - Definition of organizations managing orderer nodes
# ---------------------------------------------------------------------------
OrdererOrgs:
  # ---------------------------------------------------------------------------
  # Orderer
  # ---------------------------------------------------------------------------
  - Name: Orderer
    Domain: laser-lta.com
    EnableNodeOUs: true
    # ---------------------------------------------------------------------------
    # "Specs" - See PeerOrgs for complete description
    # ---------------------------------------------------------------------------
    Specs:
      - Hostname: orderer
        SANS:
          - localhost
```

4. Die kryptografischen Materialien mit dem Tool *cryptogen* für jede Datei im Verzeichnis *config/cryptogen* generieren

```
./bin/cryptogen generate --config=./config/cryptogen/crypto-config-manufacturer-a.yaml --
output=organizations
./bin/cryptogen generate --config=./config/cryptogen/crypto-config-user-x.yaml --
output=organizations
./bin/cryptogen generate --config=./config/cryptogen/crypto-config-orderer.yaml --
output=organizations
```

5. Die Datei *configtx.yaml* im Verzeichnis *config/configtx* erstellen, um den Kanal zu konfigurieren, der den Hersteller A, den Anwender X und den Besteller umfasst

- Die erforderlichen Ordner und die Datei erstellen

```
mkdir -p config/configtx
touch config/configtx/configtx.yaml
```

- Die folgende Konfiguration in die Datei *configtx.yaml* kopieren und einfügen

```
################################################################################
#
#   Section: Organizations
#
################################################################################
Organizations:

    - &OrdererOrg
        Name: OrdererOrg
        # ID to load the MSP definition as
        ID: OrdererMSP
        # MSPDir is the filesystem path which contains the MSP configuration
        MSPDir: ../../organizations/ordererOrganizations/laser-lta.com/msp

        # Policies defines the set of policies at this level of the config tree
        Policies:
            Readers:
                Type: Signature
                Rule: "OR('OrdererMSP.member')"
            Writers:
                Type: Signature
                Rule: "OR('OrdererMSP.member')"
            Admins:
                Type: Signature
                Rule: "OR('OrdererMSP.admin')"

        OrdererEndpoints:
            - orderer.laser-lta.com:7050

    - &ManufacturerA
        Name: ManufacturerA
        # ID to load the MSP definition as
        ID: ManufacturerAMSP
        MSPDir: ../../organizations/peerOrganizations/manufacturer-a.laser-lta.com/msp
        Policies:
            Readers:
                Type: Signature
                Rule: "OR('ManufacturerAMSP.admin', 'ManufacturerAMSP.peer',
'ManufacturerAMSP.client')"
            Writers:
                Type: Signature
                Rule: "OR('ManufacturerAMSP.admin', 'ManufacturerAMSP.client')"
            Admins:
                Type: Signature
                Rule: "OR('ManufacturerAMSP.admin')"
            Endorsement:
                Type: Signature
                Rule: "OR('ManufacturerAMSP.peer')"

    - &UserX
        Name: UserX
        # ID to load the MSP definition as
        ID: UserXMSP
        MSPDir: ../../organizations/peerOrganizations/user-x.laser-lta.com/msp
        Policies:
            Readers:
                Type: Signature
                Rule: "OR('UserXMSP.admin', 'UserXMSP.peer', 'UserXMSP.client')"
            Writers:
```

```
                Type: Signature
                Rule: "OR('UserXMSP.admin', 'UserXMSP.client')"
            Admins:
                Type: Signature
                Rule: "OR('UserXMSP.admin')"
            Endorsement:
                Type: Signature
                Rule: "OR('UserXMSP.peer')"

Capabilities:

    Channel: &ChannelCapabilities
        V2_0: true
    Orderer: &OrdererCapabilities
        V2_0: true
    Application: &ApplicationCapabilities
        V2_5: true

Application: &ApplicationDefaults

    Organizations:
    Policies:
        Readers:
            Type: ImplicitMeta
            Rule: "ANY Readers"
        Writers:
            Type: ImplicitMeta
            Rule: "ANY Writers"
        Admins:
            Type: ImplicitMeta
            Rule: "MAJORITY Admins"
        LifecycleEndorsement:
            Type: ImplicitMeta
            Rule: "MAJORITY Endorsement"
        Endorsement:
            Type: ImplicitMeta
            Rule: "MAJORITY Endorsement"

    Capabilities:
        <<: *ApplicationCapabilities

Orderer: &OrdererDefaults

    OrdererType: etcdraft
    Addresses:
        - orderer.laser-lta.com:7050
    EtcdRaft:
        Consenters:
        - Host: orderer.laser-lta.com
          Port: 7050
          ClientTLSCert: ../../organizations/ordererOrganizations/laser-
lta.com/orderers/orderer.laser-lta.com/tls/server.crt
          ServerTLSCert: ../../organizations/ordererOrganizations/laser-
lta.com/orderers/orderer.laser-lta.com/tls/server.crt

    BatchTimeout: 2s
    BatchSize:
        MaxMessageCount: 10
        AbsoluteMaxBytes: 99 MB
        PreferredMaxBytes: 512 KB

    Organizations:
    Policies:
        Readers:
            Type: ImplicitMeta
            Rule: "ANY Readers"
        Writers:
            Type: ImplicitMeta
            Rule: "ANY Writers"
        Admins:
            Type: ImplicitMeta
```

```
            Rule: "MAJORITY Admins"
        BlockValidation:
            Type: ImplicitMeta
            Rule: "ANY Writers"

Channel: &ChannelDefaults

    Policies:
        Readers:
            Type: ImplicitMeta
            Rule: "ANY Readers"
        Writers:
            Type: ImplicitMeta
            Rule: "ANY Writers"
        Admins:
            Type: ImplicitMeta
            Rule: "MAJORITY Admins"
    Capabilities:
        <<: *ChannelCapabilities


Profiles:
    ManufacturerAUserXApplicationGenesis:
        <<: *ChannelDefaults
        Orderer:
            <<: *OrdererDefaults
            Organizations:
                - *OrdererOrg
            Capabilities: *OrdererCapabilities
        Application:
            <<: *ApplicationDefaults
            Organizations:
                - *ManufacturerA
                - *UserX
            Capabilities: *ApplicationCapabilities
```

5. Netzwerk-Artefakte generierren

```
./bin/configtxgen -profile ManufacturerAUserXApplicationGenesis -outputBlock ./channel-
artifacts/a-x-channel.block -channelID a-x-channel -configPath ./config/configtx/
```

6. docker-compose-Dateien erstellen

   - die erforderlichen Dateien und Verzeichnisse ertsellen

```
mkdir -p compose/docker
touch compose/compose-net.yaml
touch compose/docker/docker-compose-net.yaml
```

   - *compose-net.yaml*-Datei kopieren und einfügen

```
nano compose/compose-net.yaml
version: '3.7'
volumes:
  orderer.laser-lta.com:
  peer0.manufacturer-a.laser-lta.com:
  peer0.user-x.laser-lta.com:
networks:
  test:
    name: fabric
services:
  orderer.laser-lta.com:
    container_name: orderer.laser-lta.com
    image: hyperledger/fabric-orderer:latest
    labels:
      service: hyperledger-fabric
    environment:
      - FABRIC_LOGGING_SPEC=INFO
      - ORDERER_GENERAL_LISTENADDRESS=0.0.0.0
```

```
      - ORDERER_GENERAL_LISTENPORT=7050
      - ORDERER_GENERAL_LOCALMSPID=OrdererMSP
      - ORDERER_GENERAL_LOCALMSPDIR=/var/hyperledger/orderer/msp
      # enabled TLS
      - ORDERER_GENERAL_TLS_ENABLED=true
      - ORDERER_GENERAL_TLS_PRIVATEKEY=/var/hyperledger/orderer/tls/server.key
      - ORDERER_GENERAL_TLS_CERTIFICATE=/var/hyperledger/orderer/tls/server.crt
      - ORDERER_GENERAL_TLS_ROOTCAS=[/var/hyperledger/orderer/tls/ca.crt]
      - ORDERER_GENERAL_CLUSTER_CLIENTCERTIFICATE=/var/hyperledger/orderer/tls/server.crt
      - ORDERER_GENERAL_CLUSTER_CLIENTPRIVATEKEY=/var/hyperledger/orderer/tls/server.key
      - ORDERER_GENERAL_CLUSTER_ROOTCAS=[/var/hyperledger/orderer/tls/ca.crt]
      - ORDERER_GENERAL_BOOTSTRAPMETHOD=none
      - ORDERER_CHANNELPARTICIPATION_ENABLED=true
      - ORDERER_ADMIN_TLS_ENABLED=true
      - ORDERER_ADMIN_TLS_CERTIFICATE=/var/hyperledger/orderer/tls/server.crt
      - ORDERER_ADMIN_TLS_PRIVATEKEY=/var/hyperledger/orderer/tls/server.key
      - ORDERER_ADMIN_TLS_ROOTCAS=[/var/hyperledger/orderer/tls/ca.crt]
      - ORDERER_ADMIN_TLS_CLIENTROOTCAS=[/var/hyperledger/orderer/tls/ca.crt]
      - ORDERER_ADMIN_LISTENADDRESS=0.0.0.0:7053
      - ORDERER_OPERATIONS_LISTENADDRESS=orderer.laser-lta.com:9443
      - ORDERER_METRICS_PROVIDER=prometheus
    working_dir: /root
    command: orderer
    volumes:
      - ../organizations/ordererOrganizations/laser-lta.com/orderers/orderer.laser-
lta.com/msp:/var/hyperledger/orderer/msp
      - ../organizations/ordererOrganizations/laser-lta.com/orderers/orderer.laser-
lta.com/tls/:/var/hyperledger/orderer/tls
      - orderer.laser-lta.com:/var/hyperledger/production/orderer
    ports:
      - 7050:7050
      - 7053:7053
      - 9443:9443
    networks:
      - test
  peer0.manufacturer-a.laser-lta.com:
    container_name: peer0.manufacturer-a.laser-lta.com
    image: hyperledger/fabric-peer:latest
    labels:
      service: hyperledger-fabric
    environment:
      - FABRIC_CFG_PATH=/etc/hyperledger/peercfg
      - FABRIC_LOGGING_SPEC=INFO
      #- FABRIC_LOGGING_SPEC=DEBUG
      - CORE_PEER_TLS_ENABLED=true
      - CORE_PEER_PROFILE_ENABLED=false
      - CORE_PEER_TLS_CERT_FILE=/etc/hyperledger/fabric/tls/server.crt
      - CORE_PEER_TLS_KEY_FILE=/etc/hyperledger/fabric/tls/server.key
      - CORE_PEER_TLS_ROOTCERT_FILE=/etc/hyperledger/fabric/tls/ca.crt
      # Peer specific variables
      - CORE_PEER_ID=peer0.manufacturer-a.laser-lta.com
      - CORE_PEER_ADDRESS=peer0.manufacturer-a.laser-lta.com:7051
      - CORE_PEER_LISTENADDRESS=0.0.0.0:7051
      - CORE_PEER_CHAINCODEADDRESS=peer0.manufacturer-a.laser-lta.com:7052
      - CORE_PEER_CHAINCODELISTENADDRESS=0.0.0.0:7052
      - CORE_PEER_GOSSIP_BOOTSTRAP=peer0.manufacturer-a.laser-lta.com:7051
      - CORE_PEER_GOSSIP_EXTERNALENDPOINT=peer0.manufacturer-a.laser-lta.com:7051
      - CORE_PEER_LOCALMSPID=ManufacturerAMSP
      - CORE_PEER_MSPCONFIGPATH=/etc/hyperledger/fabric/msp
      - CORE_OPERATIONS_LISTENADDRESS=peer0.manufacturer-a.laser-lta.com:9444
      - CORE_METRICS_PROVIDER=prometheus
      - CHAINCODE_AS_A_SERVICE_BUILDER_CONFIG={"peername":"peer0manufacturer-a"}
      - CORE_CHAINCODE_EXECUTETIMEOUT=300s
    volumes:
      - ../organizations/peerOrganizations/manufacturer-a.laser-lta.com/peers/peer0.manufacturer-
a.laser-lta.com:/etc/hyperledger/fabric
      - peer0.manufacturer-a.laser-lta.com:/var/hyperledger/production
    working_dir: /root
    command: peer node start
    ports:
      - 7051:7051
```

```
      - 9444:9444
    networks:
      - test
  peer0.user-x.laser-lta.com:
    container_name: peer0.user-x.laser-lta.com
    image: hyperledger/fabric-peer:latest
    labels:
      service: hyperledger-fabric
    environment:
      - FABRIC_CFG_PATH=/etc/hyperledger/peercfg
      - FABRIC_LOGGING_SPEC=INFO
      #- FABRIC_LOGGING_SPEC=DEBUG
      - CORE_PEER_TLS_ENABLED=true
      - CORE_PEER_PROFILE_ENABLED=false
      - CORE_PEER_TLS_CERT_FILE=/etc/hyperledger/fabric/tls/server.crt
      - CORE_PEER_TLS_KEY_FILE=/etc/hyperledger/fabric/tls/server.key
      - CORE_PEER_TLS_ROOTCERT_FILE=/etc/hyperledger/fabric/tls/ca.crt
      # Peer specific variables
      - CORE_PEER_ID=peer0.user-x.laser-lta.com
      - CORE_PEER_ADDRESS=peer0.user-x.laser-lta.com:9051
      - CORE_PEER_LISTENADDRESS=0.0.0.0:9051
      - CORE_PEER_CHAINCODEADDRESS=peer0.user-x.laser-lta.com:9052
      - CORE_PEER_CHAINCODELISTENADDRESS=0.0.0.0:9052
      - CORE_PEER_GOSSIP_EXTERNALENDPOINT=peer0.user-x.laser-lta.com:9051
      - CORE_PEER_GOSSIP_BOOTSTRAP=peer0.user-x.laser-lta.com:9051
      - CORE_PEER_LOCALMSPID=UserXMSP
      - CORE_PEER_MSPCONFIGPATH=/etc/hyperledger/fabric/msp
      - CORE_OPERATIONS_LISTENADDRESS=peer0.user-x.laser-lta.com:9445
      - CORE_METRICS_PROVIDER=prometheus
      - CHAINCODE_AS_A_SERVICE_BUILDER_CONFIG={"peername":"peer0user-x"}
      - CORE_CHAINCODE_EXECUTETIMEOUT=300s
    volumes:
      - ../organizations/peerOrganizations/user-x.laser-lta.com/peers/peer0.user-x.laser-
lta.com:/etc/hyperledger/fabric
      - peer0.user-x.laser-lta.com:/var/hyperledger/production
    working_dir: /root
    command: peer node start
    ports:
      - 9051:9051
      - 9445:9445
    networks:
      - test
  cli:
    container_name: cli
    image: hyperledger/fabric-tools:latest
    labels:
      service: hyperledger-fabric
    tty: true
    stdin_open: true
    environment:
      - GOPATH=/opt/gopath
      - FABRIC_LOGGING_SPEC=INFO
      - FABRIC_CFG_PATH=/etc/hyperledger/peercfg
      #- FABRIC_LOGGING_SPEC=DEBUG
    working_dir: /opt/gopath/src/github.com/hyperledger/fabric/peer
    command: /bin/bash
    volumes:
      - ../organizations:/opt/gopath/src/github.com/hyperledger/fabric/peer/organizations
      - ../scripts:/opt/gopath/src/github.com/hyperledger/fabric/peer/scripts/
    depends_on:
      - peer0.manufacturer-a.laser-lta.com
      - peer0.user-x.laser-lta.com
    networks:
      - test
```

- Das folgende in docker-compose-net.yaml datei kopieren und einfügen

```
nano compose/docker/docker-compose-net.yaml
version: '3.7'
services:
  peer0.manufacturer-a.laser-lta.com:
    container_name: peer0.manufacturer-a.laser-lta.com
```

```
    image: hyperledger/fabric-peer:latest
    labels:
      service: hyperledger-fabric
    environment:
      #Generic peer variables
      - CORE_VM_ENDPOINT=unix:///host/var/run/docker.sock
      - CORE_VM_DOCKER_HOSTCONFIG_NETWORKMODE=fabric
    volumes:
      - ./docker/peercfg:/etc/hyperledger/peercfg
      - /var/run/docker.sock:/host/var/run/docker.sock
  peer0.user-x.laser-lta.com:
    container_name: peer0.user-x.laser-lta.com
    image: hyperledger/fabric-peer:latest
    labels:
      service: hyperledger-fabric
    environment:
      #Generic peer variables
      - CORE_VM_ENDPOINT=unix:///host/var/run/docker.sock
      - CORE_VM_DOCKER_HOSTCONFIG_NETWORKMODE=fabric
    volumes:
      - ./docker/peercfg:/etc/hyperledger/peercfg
      - /var/run/docker.sock:/host/var/run/docker.sock
  cli:
    container_name: cli
    image: hyperledger/fabric-tools:latest
    volumes:
      - ./docker/peercfg:/etc/hyperledger/peercfg
```

7. Das Netzwerk starten

```
docker-compose -f compose/compose-net.yaml -f compose/docker/docker-compose-net.yaml up -d
```

2. Umgebungsvariablen für Config und Orderer exportieren

```
export ORDERER_CA=${PWD}/organizations/ordererOrganizations/laser-lta.com/orderers/orderer.laser-
lta.com/msp/tlscacerts/tlsca.laser-lta.com-cert.pem
export ORDERER_ADMIN_TLS_PRIVATE_KEY=${PWD}/organizations/ordererOrganizations/laser-
lta.com/orderers/orderer.laser-lta.com/tls/server.key
export ORDERER_ADMIN_TLS_SIGN_CERT=${PWD}/organizations/ordererOrganizations/laser-
lta.com/orderers/orderer.laser-lta.com/tls/server.crt
export FABRIC_CFG_PATH=$PWD/config/
export CORE_PEER_TLS_ENABLED=true
```

8. Erstellen eines Kanals mit dem Namen *a-x-channel*, um den Hersteller A und den Anwender X einzubeziehen

```
./bin/osnadmin channel join --channelID a-x-channel --config-block ./channel-artifacts/a-x-
channel.block -o localhost:7053 --ca-file "$ORDERER_CA" --client-cert
"$ORDERER_ADMIN_TLS_SIGN_CERT" --client-key "$ORDERER_ADMIN_TLS_PRIVATE_KEY"
```

9. Auflisten der Kanäle im Netzwerk, um zu bestätigen, dass der Kanal *a-x-channel* erstellt wurde

```
./bin/osnadmin channel list -o localhost:7053 --ca-file "$ORDERER_CA" --client-cert
"$ORDERER_ADMIN_TLS_SIGN_CERT" --client-key "$ORDERER_ADMIN_TLS_PRIVATE_KEY"
```

10. Erstellung eines Skripts, um die Umgebungsvariablen für eine bestimmte Organisation zu exportieren, wenn dies erforderlich ist

    - Erstellen der erforderlichen Dateien und Verzeichnisse

```
mkdir scripts
```

```
touch scripts/setOrgEnv.sh
```

- Den folgenden Text kopieren und in die Datei *setorgEnv.sh* einfügen

```
nano scripts/setorgEnv.sh
#!/bin/bash
#

# default to using ManufacturerA
ORG=${1:-ManufacturerA}

# Exit on first error, print all commands.
set -e
set -o pipefail

# Where am I?
DIR="$( cd "$( dirname "${BASH_SOURCE[0]}" )/.." && pwd )"

CORE_PEER_TLS_ENABLED=true
PEER0_ManufacturerA_CA=${PWD}/organizations/peerOrganizations/manufacturer-a.laser-
lta.com/peers/peer0.manufacturer-a.laser-lta.com/tls/ca.crt
PEER0_UserX_CA=${PWD}/organizations/peerOrganizations/user-x.laser-lta.com/peers/peer0.user-
x.laser-lta.com/tls/ca.crt

if [[ ${ORG,,} == "manufacturera" ]]; then

   CORE_PEER_LOCALMSPID=ManufacturerAMSP
   CORE_PEER_MSPCONFIGPATH=${PWD}/organizations/peerOrganizations/manufacturer-a.laser-
lta.com/users/Admin@manufacturer-a.laser-lta.com/msp
   CORE_PEER_ADDRESS=localhost:7051
   CORE_PEER_TLS_ROOTCERT_FILE=$PEER0_ManufacturerA_CA

elif [[ ${ORG,,} == "userx" ]]; then

   CORE_PEER_LOCALMSPID=UserXMSP
   CORE_PEER_MSPCONFIGPATH=${PWD}/organizations/peerOrganizations/user-x.laser-
lta.com/users/Admin@user-x.laser-lta.com/msp
   CORE_PEER_ADDRESS=localhost:9051
   CORE_PEER_TLS_ROOTCERT_FILE=$PEER0_UserX_CA

else
   echo "Unknown \"$ORG\", please choose ManufacturerA or UserX"
   echo "For example to get the environment variables to set upa UserX shell environment run:
./setOrgEnv.sh UserX"
   echo
   echo "This can be automated to set them as well with:"
   echo
   echo 'export $(./setOrgEnv.sh UserX | xargs)'
   exit 1
fi
```

3. Hersteller A mit dem Kanal verbinden

```
export $(./scripts/setOrgEnv.sh ManufacturerA | xargs)
./bin/peer channel join -b ./channel-artifacts/a-x-channel.block
```

4. Beitritt des Anwenders X zum Kanal

```
export $(./scripts/setOrgEnv.sh UserX | xargs)
./bin/peer channel join -b ./channel-artifacts/a-x-channel.block
```

11. Anker-Peer setzen

- Auswahl des Peers von Hersteller A als Anker-Peer

```
# Export environment variables
export $(./scripts/setOrgEnv.sh ManufacturerA | xargs)

# Pull the most recent channel configuration block (Because the most recent channel configuration
block is the channel genesis block, the command returns block 0 from the channel.)
./bin/peer channel fetch config channel-artifacts/config_block.pb -o localhost:7050 --
ordererTLSHostnameOverride orderer.laser-lta.com -c a-x-channel --tls --cafile "$ORDERER_CA"
```

```
# The channel configuration block config_block.pb is stored in the channel-artifacts folder
cd channel-artifacts

# Decode the block from protobuf into a JSON object
../bin/configtxlator proto_decode --input config_block.pb --type common.Block --output
config_block.json
jq '.data.data[0].payload.data.config' config_block.json > config.json

# Use the jq tool to add the ManufacturerA anchor peer to the channel configuration.
cp config.json config_copy.json
jq '.channel_group.groups.Application.groups.ManufacturerA.values +=
{"AnchorPeers":{"mod_policy": "Admins","value":{"anchor_peers": [{"host": "peer0.manufacturer-
a.laser-lta.com","port": 7051}]},"version": "0"}}' config_copy.json > modified_config.json

# Convert both the original and modified channel configurations back into protobuf format and
calculate the difference between them.
../bin/configtxlator proto_encode --input config.json --type common.Config --output config.pb
../bin/configtxlator proto_encode --input modified_config.json --type common.Config --output
modified_config.pb
../bin/configtxlator compute_update --channel_id a-x-channel --original config.pb --updated
modified_config.pb --output config_update.pb

# Wrap the configuration update in a transaction envelope to create the channel configuration
update transaction
../bin/configtxlator proto_decode --input config_update.pb --type common.ConfigUpdate --output
config_update.json
echo '{"payload":{"header":{"channel_header":{"channel_id":"a-x-channel",
"type":2}},"data":{"config_update":'$(cat config_update.json)'}}}' | jq . >
config_update_in_envelope.json
../bin/configtxlator proto_encode --input config_update_in_envelope.json --type common.Envelope -
-output config_update_in_envelope.pb

# Use the final artifact config_update_in_envelope.pb to update the channel
cd ..
./bin/peer channel update -f channel-artifacts/config_update_in_envelope.pb -c a-x-channel -o
localhost:7050 --ordererTLSHostnameOverride orderer.laser-lta.com --tls --cafile "$ORDERER_CA"
```

- Auswahl des Peers von Anwender X als Anker-Peer

```
# Export environment variables
export $(./scripts/setOrgEnv.sh UserX | xargs)

# Pull the most recent channel configuration block
./bin/peer channel fetch config channel-artifacts/config_block.pb -o localhost:7050 --
ordererTLSHostnameOverride orderer.laser-lta.com -c a-x-channel --tls --cafile "$ORDERER_CA"

# The channel configuration block config_block.pb is stored in the channel-artifacts folder
cd channel-artifacts

# Decode the block from protobuf into a JSON object
../bin/configtxlator proto_decode --input config_block.pb --type common.Block --output
config_block.json
jq '.data.data[0].payload.data.config' config_block.json > config.json
cp config.json config_copy.json

# Use the jq tool to add the UserX anchor peer to the channel configuration.
jq '.channel_group.groups.Application.groups.UserX.values += {"AnchorPeers":{"mod_policy":
"Admins","value":{"anchor_peers": [{"host": "peer0.user-x.laser-lta.com","port":
9051}]},"version": "0"}}' config_copy.json > modified_config.json

# Convert both the original and modified channel configurations back into protobuf format and
calculate the difference between them.
../bin/configtxlator proto_encode --input config.json --type common.Config --output config.pb
../bin/configtxlator proto_encode --input modified_config.json --type common.Config --output
modified_config.pb
../bin/configtxlator compute_update --channel_id a-x-channel --original config.pb --updated
modified_config.pb --output config_update.pb

# Wrap the configuration update in a transaction envelope to create the channel configuration
update transaction
```

```
../bin/configtxlator proto_decode --input config_update.pb --type common.ConfigUpdate --output
config_update.json
echo '{"payload":{"header":{"channel_header":{"channel_id":"a-x-channel",
"type":2}},"data":{"config_update":'$(cat config_update.json)'}}}' | jq . >
config_update_in_envelope.json
../bin/configtxlator proto_encode --input config_update_in_envelope.json --type common.Envelope -
-output config_update_in_envelope.pb

# Use the final artifact config_update_in_envelope.pb to update the channel
cd ..
./bin/peer channel update -f channel-artifacts/config_update_in_envelope.pb -c a-x-channel -o
localhost:7050 --ordererTLSHostnameOverride orderer.laser-lta.com --tls --cafile "$ORDERER_CA"
```

A. 2: Detaillierte Beschreibung der Implementierung des Smart Contracts

licenseModel.ts

```
import { Object, Property } from 'fabric-contract-api';

@Object()
export class OtherInformation {
    // The expiration date of the license
    @Property()
    public Expiration?: string;

    // The maximum number of times the license can be used)
    @Property()
    public NumUsages?: number;

    // A specific laser machine defined by its serial number (ID)
    @Property()
    public DeviceID?: string;

    // A specific user that can use the license defined by his email
    @Property()
    public UserEmail?: string;
}

export enum Model {
    single_hardware_time_limited = 'Single place, Hardware based, Time limited',
    single_hardware_usage_limited = 'Single place, Hardware based, Usage limited',
    multiple_user_time_limited = 'Multiple place, User based, Time limited',
    multiple_user_usage_limited = 'Multiple place, User based, Usage limited',
}

@Object()
export class LicenseModel {
    // License related information

    // Unique identifier for the license
    @Property()
    public ID: string;

    // The name of the license model
    @Property()
    public LicenseModel: Model;

    // The state of the license
    @Property()
    public Active: boolean;

    // The activation date of the license
    @Property()
    public ActivationDate: string;

    // License terms

    // other information that is used in the validation process
    @Property()
    public OtherInformation?: OtherInformation;

    // Information available after acquiring access

    // Unique identifier of the technology data file
    @Property()
    public TechnologyDataID: string;

    // The DecryptionInformation contains the symmetric key and the algorithm used to
encrypt/decrypt
    // the technology data.
```

```
    // The DecryptionInformation object is encrypted with the public key of the user and signed
by the manufacturer.
    // It is only included in the activation transaction when the license is activated.
    @Property()
    public DecryptionInformation?: string;

    // other information

    // The manufacturer that issued the license
    @Property()
    public Manufacturer: string;

    // The user that the license belongs to
    @Property()
    public User: string;
}
```

licenseContract.ts

```
import { Context, Contract, Info, Returns, Transaction } from 'fabric-contract-api';
import stringify from 'json-stringify-deterministic';
import sortKeysRecursive from 'sort-keys-recursive';
import { LicenseModel, OtherInformation, Model, LicenseActivationRecord } from './licenseModel';

@Info({ title: 'licenseContract', description: 'Smart contract for technology data licensing' })
export class LicenseContract extends Contract {

    @Transaction()
    public async InitLedger(ctx: Context): Promise<void> {
        console.info(`License Contract initialized`);
    }

    @Transaction()
    public async activateLicense(ctx: Context, id: string, licenseModel: Model, activationDate:
string, technologyDataID: string, manufacturer: string, user: string,
            decryptionInformation: string, expiration: string, numUsages: number, deviceID:
string, userEmail: string): Promise<void> {

        const exists = await this.LicenseExists(ctx, id);
        if (exists) {
            throw new Error(`The license ${id} already exists`);
        }

        if (id.endsWith('_ACTIVATION')) {
            throw new Error(`The license id ${id} is not allowed.`);
        }

        const activationTransactionID = `${id}_ACTIVATION`;
        const activationTransaction: LicenseActivationRecord = {
            ID: id,
            DecryptionInformation: decryptionInformation,
        }
        await ctx.stub.putState(activationTransactionID,
Buffer.from(stringify(sortKeysRecursive(activationTransaction))));
        // insert the license
        const otherInformation: OtherInformation = {
            Expiration: expiration !== "" ? expiration : undefined,
            NumUsages: numUsages !== -1 ? numUsages : undefined,
            DeviceID: deviceID !== "" ? deviceID : undefined,
            UserEmail: userEmail !== "" ? userEmail : undefined,
        };
        const license: LicenseModel = {
            ID: id,
            LicenseModel: licenseModel,
            Active: true,
            ActivationDate: activationDate,
            TechnologyDataID: technologyDataID,
```

```
            OtherInformation: otherInformation,
            Manufacturer: manufacturer,
            User: user,
        };
         // insert the license
        await ctx.stub.putState(id, Buffer.from(stringify(sortKeysRecursive(license))));
    }

    // ReadLicense returns the license stored with given id.
    @Transaction(false)
    public async ReadLicense(ctx: Context, id: string): Promise<string> {
        if (id.endsWith('_ACTIVATION')) {
            throw new Error(`The license ${id} does not exist`);
        }
        // get the license from chaincode state
        const licenseJSON = await ctx.stub.getState(id);
        if (!licenseJSON || licenseJSON.length === 0) {
            throw new Error(`The license ${id} does not exist`);
        }
        return licenseJSON.toString();
    }

    // get the decryption information from the activation transaction
    private async ReadDecryptionInformation(ctx: Context, id: string): Promise<string> {
        const activationTransactionID = `${id}_ACTIVATION`;
        const activationTransactionJSON = await ctx.stub.getState(activationTransactionID);
        if (!activationTransactionJSON || activationTransactionJSON.length === 0) {
            throw new Error(`The decryption information ${id} does not exist`);
        }

        const activationRecord: LicenseActivationRecord =
JSON.parse(activationTransactionJSON.toString());
        return activationRecord.DecryptionInformation;
    }

    // VerifyUser verifies that a user is authorized to use a license
    private VerifyUser(license: LicenseModel, user: string): boolean {
        return license.User === user;
    }

    // Validate expiration date
    private ValidateExpiration(license: LicenseModel): void {
        const now = new Date();
        if (license.OtherInformation  && license.OtherInformation.Expiration) {
            const expirationDate = new Date(license.OtherInformation.Expiration);
            if (now > expirationDate) {
                throw new Error(`The license ${license.ID} is expired`);
            }
        } else {
            throw new Error(`The license ${license.ID} is time-limited, but the expiration date
has not been set correctly.`);
        }
    }

    // Validate Device ID
    private ValidateDeviceID(license: LicenseModel, otherInfo: string): void {
        if (license.OtherInformation  && license.OtherInformation.DeviceID) {
            if (license.OtherInformation.DeviceID !== otherInfo) {
                throw new Error(`The license ${license.ID} is hardware-based, but an invalid
device ID was provided.`);
            }
        } else {
            throw new Error(`The license ${license.ID} is hardware-based, but the device ID has
not been set correctly.`);
        }
    }

    // Validate Num Usages
    private ValidateNumUsages(license: LicenseModel): void {
        if (license.OtherInformation  && license.OtherInformation.NumUsages >= 0) {
            if (license.OtherInformation.NumUsages === 0) {
```

```
                throw new Error(`The license ${license.ID} is usage-limited and you have exceeded
the maximum number of allowed uses.`);
            }
        } else {
            throw new Error(`The license ${license.ID} is usage-limited, but the number of usages
has not been set correctly.`);
        }
    }

    // Validate User Email
    private ValidateUserEmail(license: LicenseModel, otherInfo: string): void {
        if (license.OtherInformation  && license.OtherInformation.UserEmail) {
            if (license.OtherInformation.UserEmail !== otherInfo) {
                throw new Error(`The license ${license.ID} is user-based, but an invalid user
email was provided.`);
            }
        } else {
            throw new Error(`The license ${license.ID} is user-based, but the user email has not
been set correctly.`);
        }
    }

    // ValidateLicense verifies that a license is still active and valid
    // if the numUsages is equal to -1 then we should not check the number of usages.
    // if the expiration is empty string then we should not check the expiration date.
    private ValidateLicense(license: LicenseModel, otherInfo: string): void {
        switch (license.LicenseModel) {
            case Model.single_hardware_time_limited:
                this.ValidateExpiration(license);
                this.ValidateDeviceID(license, otherInfo);
                break;
            case Model.single_hardware_usage_limited:
                this.ValidateNumUsages(license);
                this.ValidateDeviceID(license, otherInfo);
              break;
            case Model.multiple_user_time_limited:
                this.ValidateExpiration(license);
                this.ValidateUserEmail(license, otherInfo);
                break;
            case Model.multiple_user_usage_limited:
                this.ValidateNumUsages(license);
                this.ValidateUserEmail(license, otherInfo);
                break;
            default:
                false;
                break;
          }
    }

    // AccessLicense checks if the license is still valid and
    // returns it with the decryption information after updating maxUsage
    @Transaction()
    public async AccessLicense(ctx: Context, id: string, user: string, otherInfo: string):
Promise<string> {
        const licenseJSON = await this.ReadLicense(ctx, id);
        const license: LicenseModel = JSON.parse(licenseJSON);

        // check if it is active
        if (!license.Active) {
            throw new Error(`The license ${id} is not active`);
        }

        // verify user
        const VerifyUser = this.VerifyUser(license, user);
        if (!VerifyUser) {
            throw new Error(`The user ${user} is unauthorized to access the license ${id}`);
        }

        // validate license
        try {
            this.ValidateLicense(license, otherInfo);
```

```
        } catch (error) {
            // deactivate if it is invalid
            await this.DeactivateLicense(ctx, id)
            throw new Error(`An error occurred: ${error}`);
        }

        // update NumUsages
        const updatedLicense = license;
        if (license.LicenseModel === Model.multiple_user_usage_limited || license.LicenseModel
=== Model.single_hardware_usage_limited) {
            updatedLicense.OtherInformation.NumUsages = license.OtherInformation.NumUsages - 1;
            // we insert data in alphabetic order using 'json-stringify-deterministic' and 'sort-
keys-recursive'
            await ctx.stub.putState(id,
Buffer.from(stringify(sortKeysRecursive(updatedLicense))));
        }

        // return the license with the decryption information
        updatedLicense.DecryptionInformation = await this.ReadDecryptionInformation(ctx, id);
        return stringify(sortKeysRecursive(updatedLicense));
    }

    // ReactivateLicense updates an existing license with provided parameters.
    @Transaction()
    public async ReactivateLicense(ctx: Context, id: string, activationDate: string, expiration:
string, numUsages: number, deviceID: string, userEmail: string): Promise<string> {
        const licenseJSON = await this.ReadLicense(ctx, id);
        const license: LicenseModel = JSON.parse(licenseJSON);
        license.Active = true;
        license.ActivationDate = activationDate;
        switch (license.LicenseModel) {
            case Model.single_hardware_time_limited:
                license.OtherInformation.Expiration = expiration;
                license.OtherInformation.DeviceID = deviceID;
                break;
            case Model.single_hardware_usage_limited:
                license.OtherInformation.NumUsages = numUsages;
                license.OtherInformation.DeviceID = deviceID;
              break;
            case Model.multiple_user_time_limited:
                license.OtherInformation.Expiration = expiration;
                license.OtherInformation.UserEmail = userEmail;
                break;
            case Model.multiple_user_usage_limited:
                license.OtherInformation.NumUsages = numUsages;
                license.OtherInformation.UserEmail = userEmail;
                break;
            default:
                false;
                break;
        }
        await ctx.stub.putState(id, Buffer.from(stringify(sortKeysRecursive(license))));
        return stringify(sortKeysRecursive(license));
    }
""
    // DeactivateLicense deactivates a given license.
    @Transaction()
    public async DeactivateLicense(ctx: Context, id: string): Promise<string> {
        const licenseJSON = await this.ReadLicense(ctx, id);
        const license: LicenseModel = JSON.parse(licenseJSON);
        license.Active = false;
        await ctx.stub.putState(id, Buffer.from(stringify(sortKeysRecursive(license))));
        return stringify(sortKeysRecursive(license));
    }

    // DeleteLicense deletes a given license along with its activation transaction.
    @Transaction()
    public async DeleteLicense(ctx: Context, id: string): Promise<void> {
        const exists = await this.LicenseExists(ctx, id);
        if (!exists) {
            throw new Error(`The license ${id} does not exist`);
```

```
        }
        const activationTransactionID = `${id}_ACTIVATION`;
        await ctx.stub.deleteState(activationTransactionID);
        await ctx.stub.deleteState(id);
    }

    // LicenseExists returns true when license with given ID exists.
    @Transaction(false)
    @Returns('boolean')
    public async LicenseExists(ctx: Context, id: string): Promise<boolean> {
        const licenseJSON = await ctx.stub.getState(id);
        return licenseJSON && licenseJSON.length > 0;
    }

    // GetAllLicenses returns all licenses found.
    @Transaction(false)
    @Returns('string')
    public async GetAllLicenses(ctx: Context): Promise<string> {
        const allResults = [];
        // range query with empty string for startKey and endKey does an open-ended query of all
licenses in the chaincode namespace.
        const iterator = await ctx.stub.getStateByRange('', '');
        let result = await iterator.next();
        while (!result.done) {
            // skip activation records
            if (result.value.key.includes('_ACTIVATION')) {
                result = await iterator.next();
                continue;
            }
            const strValue = Buffer.from(result.value.value.toString()).toString('utf8');
            let record;
            try {
                record = JSON.parse(strValue);
            } catch (err) {
                console.log(err);
                record = strValue;
            }
            allResults.push(record);
            result = await iterator.next();
        }
        return JSON.stringify(allResults);
    }

    // GetLicenseHistory returns a history of a license access across time
    @Transaction(false)
    @Returns('string')
    public async GetLicenseHistory(ctx: Context, id: string): Promise<string> {
        if (id.endsWith('_ACTIVATION')) {
            throw new Error(`The license ${id} does not exist`);
        }
        const exists = await this.LicenseExists(ctx, id);
        if (!exists) {
            throw new Error(`The license ${id} does not exists`);
        }
        const allResults = [];
        const iterator = await ctx.stub.getHistoryForKey(id);
        let result = await iterator.next();
        while (!result.done) {
            const strValue = Buffer.from(result.value.value.toString()).toString('utf8');
            let record;
            try {
                record = JSON.parse(strValue);
                record = {
                    ...record,
                    timestamp: result.value.timestamp.seconds
                };
            } catch (err) {
                console.log(err);
                record = strValue;
            }
            allResults.push(record);
```

```
            result = await iterator.next();
        }
        return JSON.stringify(allResults);
    }

}
```

Im Folgenden werden die Anweisungen zur Installation und zum Testen des Smart Contracts auf den Peers des Herstellers A und des Anwenders X beschrieben, die dem Kanal *a-x-channel* beigetreten sind. Es wird davon ausgegangen, dass die Smart-Contract-Dateien im Ordner *smart-contracts/license-contract* im Ordner *license-trusted-agent* sich befinden.

```
cd smart-contracts/license-contract
yarn build
cd ../..
```

1. Verpacken des intelligenten Vertrages

```
./bin/peer lifecycle chaincode package license-contract01.tar.gz --path smart-contracts/license-
contract/ --lang node --label license-contract
```

2. Einbau auf Peer 0 des Herstellers A

```
export $(./scripts/setOrgEnv.sh ManufacturerA | xargs)
./bin/peer lifecycle chaincode install license-contract01.tar.gz

# Query installed to verify installation
./bin/peer lifecycle chaincode queryinstalled --output json

# Approve the definition for manufacturer A
./bin/peer lifecycle chaincode approveformyorg -o localhost:7050 --ordererTLSHostnameOverride
orderer.laser-lta.com --tls --cafile "$ORDERER_CA" --channelID a-x-channel --name license-
contract --version 1.0 --sequence 1 --init-required

# Check whether the chaincode definition is ready to be committed (expect ManufacturerA to have
approved and UserX not to)
./bin/peer lifecycle chaincode checkcommitreadiness --channelID a-x-channel --name license-
contract --version 1.0 --sequence 1 --init-required --output json
```

3. Installation auf Peer 0 des Anwenders X

```
export $(./scripts/setOrgEnv.sh UserX | xargs)
./bin/peer lifecycle chaincode install license-contract01.tar.gz

# Query installed to verify installation
./bin/peer lifecycle chaincode queryinstalled --output json

# Approve the definition for manufacturer A
./bin/peer lifecycle chaincode approveformyorg -o localhost:7050 --ordererTLSHostnameOverride
orderer.laser-lta.com --tls --cafile "$ORDERER_CA" --channelID a-x-channel --name license-
contract --version 1.0 --sequence 1 --init-required

# Check whether the chaincode definition is ready to be committed (expect ManufacturerA and UseX
to have approved)
./bin/peer lifecycle chaincode checkcommitreadiness --channelID a-x-channel --name license-
contract --version 1.0 --sequence 1 --init-required --output json
```

4. da beide Organisationen sicher wissen, dass sie zugestimmt haben, Definitionen festlegen

```
export PEER0_ManufacturerA_CA=${PWD}/organizations/peerOrganizations/manufacturer-a.laser-
lta.com/peers/peer0.manufacturer-a.laser-lta.com/tls/ca.crt

export PEER0_UserX_CA=${PWD}/organizations/peerOrganizations/user-x.laser-
lta.com/peers/peer0.user-x.laser-lta.com/tls/ca.crt

./bin/peer lifecycle chaincode commit -o localhost:7050 --ordererTLSHostnameOverride
orderer.laser-lta.com --tls --cafile "$ORDERER_CA" --channelID a-x-channel --name license-
contract --peerAddresses localhost:7051 --tlsRootCertFiles "$PEER0_ManufacturerA_CA"  --
peerAddresses localhost:9051 --tlsRootCertFiles  "$PEER0_UserX_CA" --version 1.0 --sequence 1 --
init-required

# Confirm that the chaincode definition has been committed to the channel
./bin/peer lifecycle chaincode querycommitted --channelID a-x-channel --name license-contract --
cafile "$ORDERER_CA"
```

5. Aufrufen des Kettencodes

```
./bin/peer chaincode invoke -o localhost:7050 --ordererTLSHostnameOverride orderer.laser-lta.com
--tls --cafile "$ORDERER_CA" -C a-x-channel -n license-contract --isInit --peerAddresses
localhost:7051 --tlsRootCertFiles "$PEER0_ManufacturerA_CA" --peerAddresses localhost:9051 --
tlsRootCertFiles ${PEER0_UserX_CA} -c '{"function":"InitLedger","Args":[]}'
```

6. Smart Contract testen, indem die Funktion *GetAllLicenses* im Chaincode abgefragt wird

```
./bin/peer chaincode query -C a-x-channel -n license-contract -c '{"Args":["GetAllLicenses"]}'
```

A. 3: Detaillierte Implementierung des Frontends