

# **Eine Big-Data Architektur für die Produktion**

## **eine Annäherung aus dem Industrial Engineering**

Masterarbeit  
zur Erlangung des akademischen Grades

**Master of Science in Engineering (MSc)**

Fachhochschule Vorarlberg  
Master Informatik

Betreut von  
Mag. Dr. Peter Reiter Bakk.

Vorgelegt von  
Yannick Peter Karl Krampfl, BSc.  
Dornbirn, 29. Juli 2023

## Widmung

Ich möchte diese Seite vor Beginn meiner Masterarbeit nützen, um mich bei einigen Personen zu bedanken.

Als Erstes bedanke ich mich bei meinem Betreuer Peter Reiter (Fachhochschule Vorarlberg). Peter Reiter hat mich vor allem in der, zu anfangs schwierigen, Themenfindung unterstützt und stand mir mit seiner fachlichen Expertise stets zur Seite. Nicht nur während der Masterarbeit, sondern während des Ganzen Studiums hatte er immer ein offenes Ohr für die Studenten. Mein besonderer Dank gilt dem Partnerunternehmen dieser Arbeit, der Zumtobel Group, die gleichzeitig mein Arbeitgeber ist. Ich bedanke mich für die Freiheiten in der Umsetzung und das Vertrauen, dass ich während der Entwicklung des Konzeptes genoss. Spezieller Dank gebührt an dieser Stelle meinen Vorgesetzten und Kollegen, die meine Ideen stets unterstützt haben, es ist eine Bereicherung, in eurem Team arbeiten zu dürfen.

Eine besondere Stellung nehmen meine Freunde und meine Familie ein, allen voran aber meine Ehefrau, die mir stets den Rücken freigehalten hat. Ich bin zutiefst dankbar für eure Unterstützung, vor allem für euer Verständnis gegenüber meiner Abwesenheit, da das Verfassen dieser Arbeit in den letzten Monaten den Großteil meiner Aufmerksamkeit beansprucht hat. Ohne euch wäre die Fertigstellung dieser Masterarbeit nicht möglich gewesen.

*„Erfolg ist, von Niederlage zu Niederlage zu gehen, ohne seine Begeisterung zu verlieren.“ - Winston Churchill*

# Kurzreferat

## **Eine Big-Data Architektur für die Produktion - eine Annäherung aus dem Industrial Engineering**

Diese Masterarbeit untersucht die Entwicklung einer Big-Data Architektur speziell für den Produktionsbereich. Angesichts des rasanten Anstiegs der anfallenden Datenmengen in produzierenden Unternehmen und der steigenden Nachfrage nach datenbasierten Erkenntnissen ist eine effektive Big-Data Architektur von entscheidender Bedeutung. Speziell in Hochlohnländern ist es notwendig für die Betriebe, sich mit modernen Technologien einen Wettbewerbsvorteil zu verschaffen. Vor allem der Fachbereich des Industrial Engineering ist für Produktions- und Prozessoptimierung sowie die Einführung neuer Technologien, oft die erste Anlaufstelle. Das Ziel dieser Arbeit ist es, die Herausforderungen und Anforderungen an Big-Data Anwendungen, aus der Sicht des Industrial Engineering, in der Produktion durchzuführen und darauf aufbauend eine Architektur zu entwerfen. Zunächst erfolgt eine umfassende Literaturrecherche, um einen Überblick über vorhandene Big-Data Architekturen und deren Anwendungen in der Produktion zu erhalten. Die identifizierten Anforderungen, wie etwa die Verarbeitung großer Datenmengen in Echtzeit, die Integration verschiedener Datenquellen und die Gewährleistung von Skalierbarkeit, werden detailliert analysiert. Anschließend werden die Anforderungen in einem Proof-Of-Concept umgesetzt. Dies geschieht mithilfe der Implementierung eines Prototyps einer Big-Data-Architektur. Die Architektur wird im Produktionsumfeld, anhand eines konkreten Anwendungsfalles getestet und im Anschluss evaluiert und analysiert. Basierend auf den Erkenntnissen der Literaturrecherche und der Analyse des Prototyps wird ein Handlungsleitfaden abgeleitet, der bei der zukünftigen Umsetzung von Big-Data Projekten als technologische und organisatorische Orientierungshilfe dienen soll.

# Abstract

## **A Big-Data architecture for production - an industrial engineering approach**

This master thesis investigates the development of a Big-Data architecture specifically for the manufacturing sector. In view of the rapid increase in the amount of data in manufacturing companies and the increasing demand for data-based insights, an effective Big-Data architecture is crucial. Especially in high-wage countries, it is necessary for companies to gain a competitive advantage with modern technologies. Especially the Industrial Engineering department is for production and process optimization and the introduction of new technologies, often the first point of contact. The goal of this thesis is to analyze the challenges and requirements in the area of Big-Data from the point of view of industrial engineering in production and to design an architecture based on this. architecture. First, a comprehensive literature review is conducted to obtain an overview of existing Big-Data architectures and their applications in production. The identified requirements, such as processing large amounts of data in real time, integrating different data sources and ensuring scalability, are analyzed in detail. Subsequently, the requirements are implemented in a proof-of-concept. This is done with the help of the implementation of a prototype Big-Data architecture. The architecture is tested in the production environment, based on a concrete use case, and subsequently evaluated and analyzed. Based on the findings of the literature research and the analysis of the prototype a guideline for action will be derived that will serve as a technological and organizational orientation for the future implementation of Big-Data projects.

# Inhaltsverzeichnis

<b>Abbildungsverzeichnis</b>	<b>7</b>
<b>Abkürzungsverzeichnis</b>	<b>8</b>
<b>1 Einleitung</b>	<b>9</b>
1.1 Zielsetzung . . . . .	10
1.2 Aufbau der Arbeit . . . . .	10
<b>2 Anwendungsbereich</b>	<b>11</b>
2.1 Industrie 4.0 . . . . .	12
2.1.1 Cyber-Physikalische-Systeme . . . . .	13
2.1.2 Internet der Dinge . . . . .	14
2.1.3 Smart Factory . . . . .	14
2.1.4 Vertikale Integration . . . . .	15
2.1.5 Horizontale Integration . . . . .	15
2.1.6 Durchgängiges Engineering . . . . .	17
2.2 Big-Data . . . . .	18
2.3 Methodische Vorgehensweise . . . . .	21
2.3.1 Qualitative Inhaltsanalyse . . . . .	21
2.3.2 Interpretative Vorgehensweise . . . . .	22
2.3.3 Recherche Ergebnisse . . . . .	24
2.3.4 Zusammenfassung der Ergebnisse . . . . .	31
<b>3 Proof-Of-Concept</b>	<b>32</b>
3.1 Vorgehensweise . . . . .	32
3.2 Ausgangssituation . . . . .	37
3.2.1 Tecton Produktbeschreibung . . . . .	37
3.2.2 Tecton Montageprozess . . . . .	38
3.3 Die Referenzarchitektur . . . . .	41
3.3.1 Data Providers . . . . .	42
3.3.2 Data Layer . . . . .	42
3.3.3 Read-Only Datastores . . . . .	43
3.3.4 API Architecture . . . . .	44
3.3.5 Streaming Architecture . . . . .	44

3.3.6	Event Driven Architecture . . . . .	45
3.3.7	Event Sourcing . . . . .	45
3.3.8	Data Consumers . . . . .	47
3.4	Umsetzung der Big-Data-Architektur . . . . .	48
3.4.1	Systemgrenzen . . . . .	48
3.4.2	Prototyp Architektur . . . . .	49
3.5	Analyse . . . . .	58
<b>4</b>	<b>Abgeleiteter Handlungsleitfaden</b>	<b>62</b>
<b>5</b>	<b>Zusammenfassung und Ausblick</b>	<b>69</b>
5.1	Zusammenfassung . . . . .	69
5.2	Fazit . . . . .	71
5.3	Ausblick . . . . .	74
	<b>Literaturverzeichnis</b>	<b>75</b>
	<b>Anhang</b>	<b>80</b>
	<b>Eidesstattliche Erklärung</b>	<b>83</b>

# Abbildungsverzeichnis

2.1	Horizontale/Vertikale Integration . . . . .	17
2.2	Eingesetzte Technologien in Unternehmen im Jahr 2018 . . . . .	18
2.3	Eingesetzte Technologien im Jahr 2022 . . . . .	19
2.4	Prognose weltweit generiertes Datenvolumen bis 2025 . . . . .	20
2.5	Ablaufmodell induktive Kategorienbildung . . . . .	24
2.6	Kategorienbaum Struktur Übersicht . . . . .	25
2.7	Kategorienbaum Relevante Daten . . . . .	25
2.8	Kategorienbaum Potentiale und Bedeutung von Big Data in der Produktion . . . . .	26
2.9	Kategorienbaum Anforderungen an Big Data Architekturen . . . . .	28
2.10	Kategorienbaum Herausforderung in der Implementierung . . . . .	30
3.1	DSRM Prozess . . . . .	33
3.2	DSRM angewandt . . . . .	36
3.3	Querschnitt Tecton Tragschiene . . . . .	38
3.4	Tecton Anlagenslayout . . . . .	39
3.5	Prägung TECTON Tragschiene . . . . .	40
3.6	Architektur Ausgangssituation . . . . .	41
3.7	Referenzarchitektur Data Management At Scale . . . . .	42
3.8	Event Sourcing Schema . . . . .	46
3.9	Event Routing in der Cloud . . . . .	47
3.10	System Übersicht . . . . .	49
3.11	Prototyp Architektur . . . . .	50
3.12	MQTT Prinzip . . . . .	52
3.13	Dataframe Stream . . . . .	54
3.14	Dataframe Transformiert . . . . .	54
3.15	Vergleich Prototypen . . . . .	59
3.16	Erweiterte Architektur . . . . .	61
4.1	Handlungsleitfaden . . . . .	68

# Abkürzungsverzeichnis

**API** Application Programmable Interface

**CAD** Computer-Aided-Design

**CSV** Comma-Separated-Values

**CQRS** Command Query Responsibility Segregation

**ETL** Extract Transform Load

**IPC** Industrial-Personal-Computer

**IOT** Internet-Of-Things

**JSON** Javascript Objective Notation

**MQTT** Message Queuing Telemetry Transport

**POC** Proof-Of-Concept

**RDS** Read-Only-Datastore

**RFID** Radio-Frequency-Identification

**SAS** Shared-Access-Signature

**SPS** Speicherprogrammierbare Steuerung

**XML** Extensible Markup Language



# 1 Einleitung

Diese Masterarbeit beschäftigt sich mit der Umsetzung einer skalierbaren und möglichst universell anwendbaren Big-Data Architektur für produzierende Unternehmen. Das produzierende Gewerbe hat in der Regel sehr spezielle, branchenabhängige Ziele zur Nutzung von Big-Data Plattformen und bietet deshalb interessante Anwendungsfälle. Die vorliegende Arbeit wurde in Zusammenarbeit mit der Zumtobel Group, mit Sitz in Dornbirn, Österreich, verfasst. Die Zumtobel Group ist ein weltweit agierender Leuchtenhersteller im Premium Segment, dessen Produktion in Österreich einen hohen Automatisierungsgrad aufweist. In vielen produzierenden Unternehmen werden die Potenziale, die Big-Data Methoden bieten, nicht genutzt. Das liegt teilweise an fehlendem Wissen, kann aber auch am Mangel an Verständnis oder Akzeptanz für die Thematik liegen. Um in Mitteleuropa weiterhin wettbewerbsfähig produzieren zu können, kann die Anwendung dieser Technologien ein Treiber für Erfolg sein. Die Verschmelzung von Produktionstechnik mit IT-Technologien wird in der Industrie und der Forschung meistens dem Begriff Industrie 4.0 zugeordnet, aus diesem Grund bedient sich auch diese Arbeit einiger Begrifflichkeiten aus diesem Fachbereich. Industrie 4.0 - Technologien unterliegen einem ständigen Wandel, aber die grundlegenden Anforderungen der Datenproduzenten und Datenkonsumenten, wie Datenversand und Datentransformation, bleiben bestehen. Deshalb benötigen Unternehmen eine skalierbare, modulare und flexible Architektur, die Datenerzeuger und Nutzer nahtlos miteinander verbindet und Transparenz über die Prozesse gewährt. Eine solche Architektur wird als skalierbar bezeichnet.[1] Das Industrial Engineering als Fachbereich, ist in vielen Unternehmen ein Service Dienstleister in Produktionsfragen und Werksplanung. Das Entwickeln und Optimieren von Prozessen im Produktionsumfeld ist oft, so auch bei der Zumtobel Group, in diesem Bereich angesiedelt. Deshalb ist das Industrial Engineering oft die erste Anlaufstelle für die Umsetzung von Technologieprojekten. Die steigende Relevanz von Big-Data in der Fertigung stellt auch das Industrial Engineering vor neue Herausforderungen, da die Angestellten meistens Ingenieure aus Maschinen- oder Anlagenbau, und somit Domänen Fremde sind. Speziell bei der Zumtobel Group ist eine Tendenz in diese Richtung zu beobachten, deswegen wird diese Arbeit aus der Sicht des Industrial Engineers betrachtet. Derzeit gibt es nur einen Anwendungsfall im Unternehmen, bei dem gezielt Daten aus der Produktion gesammelt werden, dieser erfüllt je-

doch nicht die wichtigsten Grundvoraussetzungen wie etwa Modularität und Skalierbarkeit. Zusammen mit dem Partnerunternehmen soll ein Konzept für die Nutzung von Big-Data in der Produktion ausgearbeitet und mit einer prototypischen Implementierung getestet werden. Die Implementierung dient als Proof-Of-Concept für die Evaluierung der recherchierten Anforderungen. Abschließend wird aus den Erfahrungen aus Recherche und Implementierung ein Handlungsleitfaden abgeleitet, der für die zukünftige Umsetzung von Big-Data Projekten, speziell aus Sicht des Industrial Engineering, herangezogen werden kann.

## 1.1 Zielsetzung

Die Zielsetzung der Arbeit wird mit den folgenden drei Punkten beschrieben:

- Ausarbeiten von Anforderungen an Big-Data Architekturen in der Produktion
- Entwicklung eines Prototyps, der die Anforderungen erfüllt
- Entwicklung eines Handlungsleitfadens für die Umsetzung zukünftiger Big-Data-Projekte

## 1.2 Aufbau der Arbeit

Am Anfang der Arbeit wird kurz auf den Anwendungsbereich der Masterarbeit im Umfeld von Industrie 4.0, Produktion und Big-Data eingegangen. Kapitel 2 gibt einen Überblick über die wichtigsten Begriffe von Industrie 4.0 und eine Einführung in den Begriff „Big-Data“. Anschließend folgt eine Literaturrecherche, deren Ergebnisse mittels einer qualitativen Inhaltsanalyse, bewertet werden. Basierend auf diesen Ergebnissen werden die Anforderungen, Herausforderungen und Potenziale der Nutzung von Big-Data in der Produktion sowie die Daten, die für die Umsetzung einer Big-Data Architektur relevant sind, erhoben. Die Ergebnisse der Recherche bilden die Grundlage für die Entwicklung einer prototypischen Big-Data Architektur in Kapitel 3.4.2. Der Prototyp dient als Proof-Of-Concept dessen Entwicklung sich zielorientiert, nach der „Design Science for Information Systems Research“ Methode richtet. Aus den Ergebnissen des Proof-Of-Concept und der Literaturrecherche entsteht ein Handlungsleitfaden, der bei der Implementierung einer Big-Data Architektur zu Rate gezogen werden kann. Im letzten Kapitel werden die Ergebnisse analysiert, zusammengefasst und es wird ein Ausblick auf die Zukunft gegeben.

## 2 Anwendungsbereich

Der Anwendungsbereich dieser Arbeit liegt im Umfeld eines europäischen Produktionsunternehmens mit hohem bis steigendem Automatisierungsgrad. Da Produktionsunternehmen in Hochlohnländern stetigem Wettbewerbsdruck ausgesetzt sind, kommt es vermehrt zu Verlagerungen ganzer Produktionsstätten in Niedriglohnländer. Unternehmen stehen vor unterschiedlichen Herausforderungen wie: schnell, termingerecht und flexibel fertigen und ausliefern zu können. Dabei wird von Kunden hohe Qualität vorausgesetzt, genauso wie die Nachverfolgbarkeit von Komponenten im Schadensfall. Die Produkte werden „intelligenter“, „individueller“ und sollen zu üblichen Preisen auf dem Markt verfügbar sein. Bei Produktionsverlagerungen werden nicht nur wertschöpfende Tätigkeiten transferiert, sondern auch Technologien, Entwicklungsarbeit und somit Wissen. Auf lange Sicht verbessern Produzenten in Niedriglohnländern ihre technischen Fähigkeiten und werden selbst zu Wissensträgern und Innovationstreibern. Dadurch nimmt der Druck auf die europäische Industrie weiter zu.[2, S.18-22] Um sicherzustellen, als Unternehmen in einem Hochlohnland wettbewerbsfähig zu bleiben, gibt es unterschiedliche Ansätze. Einer davon ist die sukzessive Automatisierung von Prozessen, die früher „manuell“ durchgeführt wurden und deshalb hohe Lohnkosten verursacht haben. Automatisierung geht einher mit hohen Investitionskosten und einem nicht unerheblichen Wartungsaufwand. Um die automatisierten Prozesse zu kontrollieren, die Fertigung termingerecht planen zu können und Ausfälle frühzeitig zu prognostizieren, ist es hilfreich, die Produktion zu digitalisieren. Digitalisierung heißt in diesem Zusammenhang, das Schaffen von Transparenz über die Prozesse durch: Erheben, Analysieren und Verarbeiten der anfallenden Prozessdaten. Somit werden nicht nur die Produkte, sondern auch die notwendigen Maschinen und Anlagen intelligenter. Einer der Haupttreiber dieser Umwälzungen ist das „Internet der Dinge“. Das Internet der Dinge besteht aus autonomen, eingebetteten Systemen, die Daten erfassen und miteinander über drahtlose Netzwerke kommunizieren. Dadurch entstehen sogenannte „Cyber-Physikalische-Systeme“ (CPS). Diese Systeme werden durch ihre Eigenschaften wie Programmierbarkeit, Kommunikationsfähigkeit und Speichervermögen intelligent. Die Wissenschaft und Industrie spricht von „Industrie 4.0“ oder der „vierten industriellen Revolution“. Die von den CPS generierten, riesigen Datenmengen werden oft unter dem Begriff „Big-Data“ zusammengefasst und dienen als Enabler für die Moderni-

sierung von industriellen Prozessen.[3]

In diesem Kapitel werden die Begriffe Industrie 4.0 und Big-Data näher erläutert und im Kontext der Produktion und Automatisierung betrachtet. Danach werden die Anforderungen, die die Industrie an Big-Data Systemarchitekturen stellt, aus einer Literaturrecherche gewonnen und beschrieben. Im Anschluss daran werden einige bekannte Herausforderungen in der Implementierung einer Big-Data Architektur in der Produktion diskutiert. Die Anforderungen und Herausforderungen aus Kapitel 2.3.3 sind vital für die Ausarbeitung des Prototyps in Kapitel 3.4 und den abgeleiteten Handlungsleitfaden in Kapitel 4.

## 2.1 Industrie 4.0

Experten sprechen von der vierten Industriellen Revolution, oder einer „Evolution“ in der die reale und die virtuelle Welt miteinander verschmelzen. Neue Produktionsmethoden und Prozesse werden dadurch denkbar. Nicht nur Maschinen und integrierte Systeme kommunizieren miteinander, sondern alle Systeme sind intelligent miteinander vernetzt und tauschen in Echtzeit Informationen miteinander aus [4, S.15]. Bekannte Produktionsprozesse werden durch den Einsatz moderner Informationstechnologie „intelligenter“. Durch die Verschmelzung von realer und virtueller Welt entstehen sogenannte Cyber-Physical-Systems (CPS) die mithilfe des Internet-of-Things (IoT) miteinander kommunizieren und die Basis der „Smart Factory“ bilden. Diese Anwendungen entwickeln sich stetig weiter und erzeugen Unmengen von Daten, die gespeichert und verarbeitet werden müssen. Zu diesem Zweck werden Daten-Architekturen bereitgestellt, die auf lokalen Servern oder in der Cloud arbeiten.[3] Ein Beispiel für CPS sind Sensoren an Maschinenstationen die physikalische Veränderungen wie elektrische Leistungsaufnahme, Druckveränderungen und Temperaturen an eine Steuerung melden. Die Steuerung analysiert diese Veränderungen und kann auf Fehlverhalten einzelner Komponenten aufmerksam machen. Unternehmen, die eine Vernetzung dieser Art vorantreiben, bringt das, Kosten, Zeit und Effizienzvorteile. Sie können somit ihre Wettbewerbsfähigkeit erhalten und gegebenenfalls steigern. Industrie 4.0 erfordert ein umfassendes Management, das auf kontinuierlicher Echtzeit-Verfügbarkeit der relevanten Daten, in der richtigen Qualität und der Vernetzung aller an der Produktion beteiligten Einheiten basiert [5, S.17]. Um ein Industrie 4.0 System einzuführen, ist eine flächendeckende Nutzung von Echtzeit – Informationen notwendig. Die Implementation geschieht anhand der folgenden drei Kernelemente von Industrie 4.0:

- Horizontale Integration: Netzwerke entstehen über die Unternehmensgrenzen hinaus

- Vertikale Integration: Innerhalb eines Unternehmens werden flexible und rekonfigurierbare Produktionssysteme aufgebaut
- Durchgängiges Engineering: Interdisziplinäres, ganzheitliches Engineering über die gesamte Wertschöpfungskette eines Produktes und die Produktionssysteme

Industrie 4.0 ist ein Konzept, das nur mithilfe einiger Technologien funktioniert und umgesetzt werden kann. Diese Technologien werden in den folgenden Unterkapiteln erklärt.

### 2.1.1 Cyber-Physikalische-Systeme

Cyber-Physikalische-Systeme sind ein wichtiger Bestandteil aller Industrie 4.0 Anwendungen. CPS sind komplexe Systeme bestehend aus einem Verbund von mechanischen, elektronischen und Software-Komponenten. Der Grundgedanke hinter CPS ist es mechanische Teile durch Sensoren, Aktoren und Mikrocontroller intelligent zu gestalten [6, S.15]. Eine Kerneigenschaft von CPS ist die Kommunikationsfähigkeit mit anderen Systemen. Erst dadurch werden sie zu vollwertigen CPS und erst durch diese Fähigkeit kann das volle Potenzial dieser Technologie ausgeschöpft werden. CPS verfügen neben ihren physischen Eigenschaften auch über eine virtuelle Komponente, die mittels eines Datenmodells den Zustand des Systems digital beschreibt.[7, S.39] Wenn CPS im Umfeld von Produktionen verwendet werden, bezeichnet man sie oft als Cyber-Physische-Produktions-Systeme (CPPS). CPPS bilden ein Netzwerk von miteinander verbundenen Produktionssystemen, deren Haupteigenschaften Adaptivität, Flexibilität und Effizienz sind. [7, S.40] CPPS können Daten erfassen, analysieren und auf deren Basis Entscheidungen treffen. Dabei handelt es sich bei CPPS nicht um vollwertige Maschinen und Anlagen, sondern lediglich um Komponenten wie Sensoren, Aktoren und Mikrocontroller. In der Realität werden nicht alle Sensoren und Aktoren über die oben genannten Eigenschaften verfügen. Wenn die Systeme aber in einem IT-System verwaltet werden, und die Fähigkeit haben, mit anderen Systemen zu kommunizieren werden sie als vollwertige CPPS angesehen. Oft ist es praktikabel, gewisse Funktionen eines CPPS in ein übergeordnetes IT-System auszulagern [7, S.41]. In Smart Factories sind CPPS ein wichtiger Bestandteil für die Prozess- und Produktionssteuerung. Dazu stellen CPPS die gesammelten Daten über das Internet zur Verfügung, wo sie von der Außenwelt genutzt werden können. Mithilfe dieser Eigenschaft bilden CPPS den Übergang von der realen in die virtuelle Welt und es entsteht das Internet der Dinge [8]. Die Ausführungen von CPPS können vielfältig sein, ein klassisches Beispiel aus dem Umfeld der Produktion ist ein Werkstückträger in einer automatischen Anlage. Der Werkstückträger kann mit einem RFID-Transponder

ausgestattet und eindeutig identifiziert werden, permanent werden Daten über seine Position und seinen Zustand gesammelt (trägt Werkstück oder nicht) und über WLAN, in einem standardisierten Protokoll an einen Empfänger gesendet [5, S.19]. Nicht nur Produktionsmittel, sondern auch Produkte können CPS sein. Durch den Einsatz von Industrie 4.0-Technologien, werden auch die Produkte quasi „intelligent“. Ein häufiger Anwendungsfall hierfür ist die eindeutige Kodierung von verbauten Komponenten, zum Beispiel durch QR-Codes. Dadurch kann ein Bauteil innerhalb eines Produktes von der Produktion bis zum Endkunden nachverfolgt werden. Viele moderne Produkte enthalten ebenfalls eingebettete Systeme zur Steuerung. Diese eingebetteten Systeme können Daten sammeln und diese zur weiteren Verwendung zur Verfügung stellen.

### **2.1.2 Internet der Dinge**

In der Fachliteratur und Industrie wird für das Internet der Dinge oft die englische Abkürzung „IoT“ verwendet, in dieser Arbeit kommen beide Begriffe vor und werden synonym verwendet. Seit das Internet die Industrie und die privaten Haushalte erobert hat, passen sich die sozialen und wirtschaftlichen Strukturen, in denen wir leben, stetig an. Durch die zunehmende Vernetzung von Geräten, Maschinen und Sensoren über das Internet entsteht das „Internet der Dinge“, in dem die virtuelle und physische Welt ineinander übergehen [3, S.18]. Das Internet der Dinge kann als die Infrastruktur für den Informationsaustausch zwischen CPS gesehen werden. Vier wichtige Grundideen des Internets der Dinge werden von HBK wie folgt definiert [9]:

- individuelle Informationen werden direkt am Objekt gespeichert
- Vernetzung der Objekte untereinander über das Internet
- Treffen individueller Entscheidungen auf Basis lokal oder zentral ausgewerteter Datenströme (Mithilfe von Big-Data Methoden)
- ereignisorientierte Steuerung von Prozessen in Echtzeit

Das Internet der Dinge ist jener Bereich der Industrie 4.0 in dem die meisten Daten anfallen, die verarbeitet werden müssen. Aus diesem Grund spielt Datenmanagement mit Big-Data Methoden eine besondere Rolle in der modernen Produktion [3, S.18].

### **2.1.3 Smart Factory**

Eine Smart Factory, ist ein hochgradig vernetztes und automatisiertes Produktionsumfeld, das die zuvor genannten Technologien wie das Internet der Dinge

(IoT), Cyber-Physische-Produktionssysteme (CPPS) und Big-Data integriert. Das grundlegende Ziel einer Smart Factory, ist das Sammeln und Auswerten von Informationen aus der laufenden Produktion, um Transparenz über die Prozesse zu schaffen.[10] Die Smart Factory ist Teil einer horizontalen Integration über die Wertschöpfungskette, ihre Produktionssysteme sind vertikal vernetzt und sie weist ein durchgängiges Engineering auf.[3, S.12] Aus der oben ausgearbeiteten Definition lässt sich ableiten, welche Themengebiete in einer Smart Factory in finaler Ausbaustufe auftreten:

- Vertikale Integration
- Horizontale Integration
- Vernetzung von CPS zum Internet der Dinge
- Big-Data

Nimmt man nun intelligente und kommunikationsfähige Produktionssysteme, lässt diese Daten erheben und vernetzt sie über das Internet der Dinge, vertikal und horizontal miteinander, entsteht eine Smart Factory.

#### **2.1.4 Vertikale Integration**

Die vertikale Integration beschreibt die Vernetzung innerhalb eines Unternehmens entlang seiner Organisationsebenen, von Managementebene bis zur Produktion. Daten sollen der Fertigungsplanung in Echtzeit zur Verfügung gestellt werden, um bei der Steuerung des Produktionsbetriebes und der Entscheidungsfindung zu unterstützen.[11] Die von CPS gesammelten Daten müssen auf einer gemeinsam nutzbaren Plattform, zum Beispiel einer Cloud, bereitgestellt werden. Für die Realisierung ist es notwendig, standardisierte Lösungen bezüglich der Datenqualität und Datenübertragungsprotokolle von Sensoren und Aktoren einzuführen.[12]

#### **2.1.5 Horizontale Integration**

Im Gegensatz zur vertikalen Integration spielt bei der horizontalen Integration die Vernetzung entlang des Prozesses eine Rolle. Die horizontale Integration enthält Elemente der digitalen Transformation, da sie voraussetzt, dass Silos aufgelöst und Abteilungen miteinander verknüpft werden. [13] Horizontale Integration bedeutet, dass alle Bereiche eines Unternehmens von Produktentwicklung bis zu Verkauf und Service miteinander über ein IT-System verbunden sind. Rein auf die Produktion bezogen bedeutet horizontale Integration die

Vernetzung von Maschinen und Anlagen zu einem Netzwerk von CPPS.[14] Informationen sollen dabei durchgängig fließen, möglich ist sogar eine horizontale Integration über die Unternehmensgrenzen hinaus.[3, S.24] Ein bisher oft praktizierter Anwendungsfall ist die direkte Anbindung von Lieferanten an die Lagerverwaltung. Dabei werden Lieferanten direkt an ein Meldesystem angeschlossen, welches automatisch eine Bestellung auslöst, sobald ein bestimmter Bestand erreicht oder unterschritten wird. Eine andere moderne Herangehensweise bietet die Firma TRUMPF mit ihrem „Social Machines“ Ansatz. Bei diesem Konzept sind Fertigungsmaschinen untereinander vernetzt. Sobald der Kunde ein neues Material oder Verfahren auf einer Maschine starten möchte, lädt die Maschine von einem übergeordnetem System oder anderen Maschinen die notwendigen Parameter.[5, S.29] Durch diese Vorgehensweise der horizontalen Integration entstehen neue Produktionsnetzwerke und Geschäftsfelder.[3] Die Realisierung eines derartigen Netzwerkes wird durch den Einsatz von CPS und IoT-Schnittstellen möglich. Eine potenzielle Anwendung der Zukunft könnte die direkte Anbindung von Maschinen und Anlagen bei Kunden an eine Wissensplattform sein, nun kommen Big-Data Konzepte ins Spiel. So kann Wissen nicht nur nachgeschlagen und wiederverwendet, sondern auch gespeichert und unter Anwendung von Machine Learning Algorithmen selbständig erlernt werden. Abbildung 2.1 stellt die horizontale und vertikale Integration innerhalb eines Unternehmens dar.



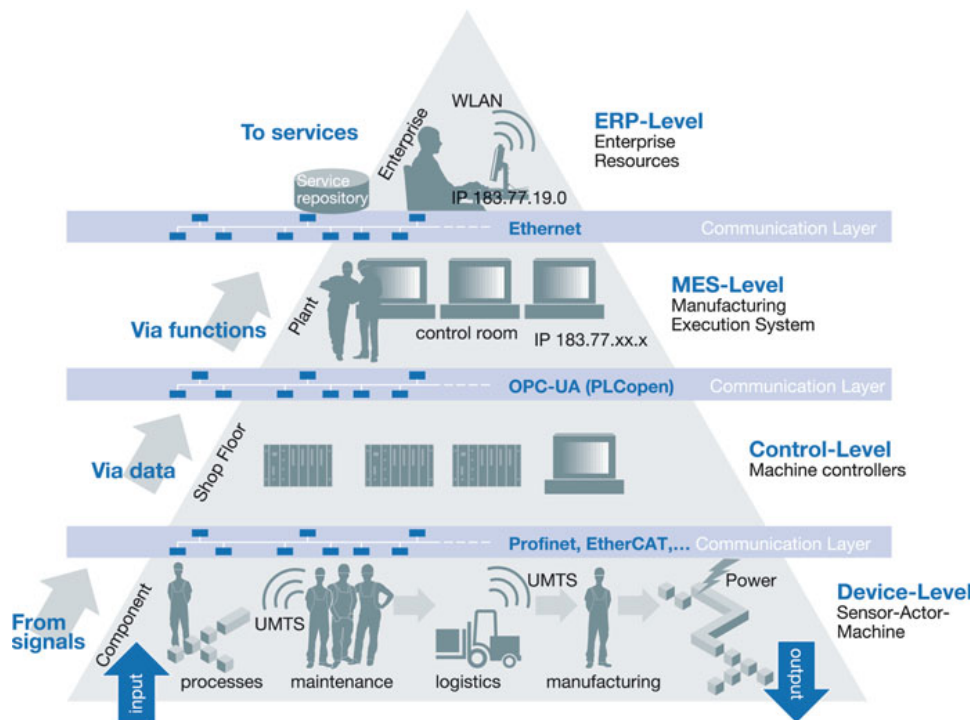


Abbildung 2.1: Horizontale/Vertikale Integration  
Quelle: OPC, entnommen aus[15]

## 2.1.6 Durchgängiges Engineering

Beim durchgängigen Engineering geht es wie bei der vertikalen und horizontalen Integration darum, einen ganzheitlichen Informationsfluss zu schaffen. Die gesamte Wertschöpfungskette von Produktentwicklung bis Produktion soll digitalisiert sein, das heißt, dass Datenmodelle von Produkten über den gesamten Produktlebenszyklus gepflegt werden. Datenmodelle bedeuten in diesem Zusammenhang das Vorhandensein von „digitalen Zwillingen“ sowohl von Produkten als auch von Cyber-Physischen-Systemen und Anlagen.[3] Das durchgängige Engineering betrifft somit sowohl die vertikale als auch die horizontale Integration, da sie sicherstellen muss, dass die Datenmodelle zusammenpassen und ein gemeinsames Verständnis zwischen den Ebenen und Systemen herrscht.[7]

## 2.2 Big-Data

Durch den Aufbau einer durchgängigen und digitalisierten Unternehmensstruktur in vertikaler und horizontaler Richtung entstehen große Datenströme, die gesteuert, gespeichert, analysiert und verwaltet werden müssen. Eine Umfrage von Statista aus dem Jahr 2018, stellte der Industrie folgende Frage:

- Welche der folgenden innovativen Anwendungen setzen Sie aktuell ein oder planen Sie einzusetzen?

Nach den Ergebnissen der Umfrage belegte Big-Data den fünften Platz, hinter Cloud-Computing, IoT, Smart Devices, Robotik und Sensorik und 3D Druck. Die Auswertung dieser Umfrage ist in Abbildung 2.2 ersichtlich.

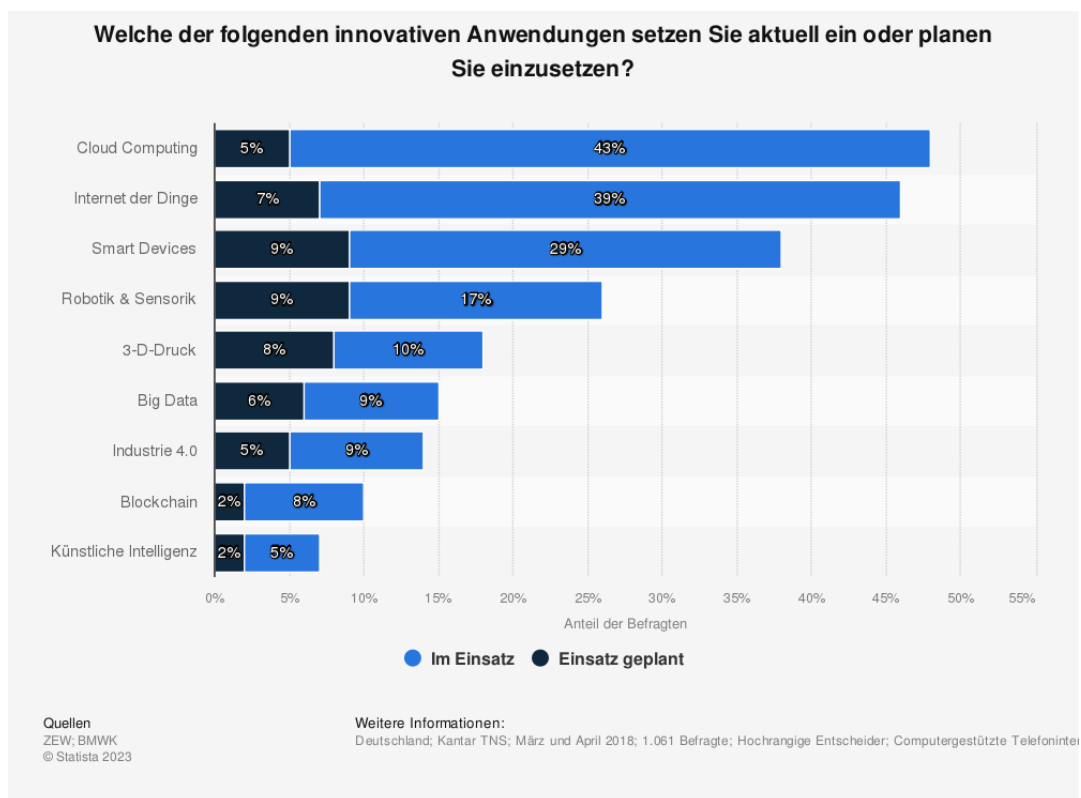


Abbildung 2.2: Eingesetzte Technologien in Unternehmen im Jahr 2018

Quelle: Statista 2018:

Eine aktuellere Umfrage aus dem Jahr 2022 stellte der Industrie die Frage:

- Welche Technologien werden in Ihrem Unternehmen eingesetzt oder sollen eingesetzt werden?

Die Auswertung zeigt, dass Big-Data inzwischen auf Platz eins, vor den zuvor genannten Technologien liegt. Folglich hat das Interesse der Industrie, an Big-Data, in den vergangenen fünf Jahren deutlich zugenommen. Unter Abbildung 2.3 ist die Statistik grafisch dargestellt.

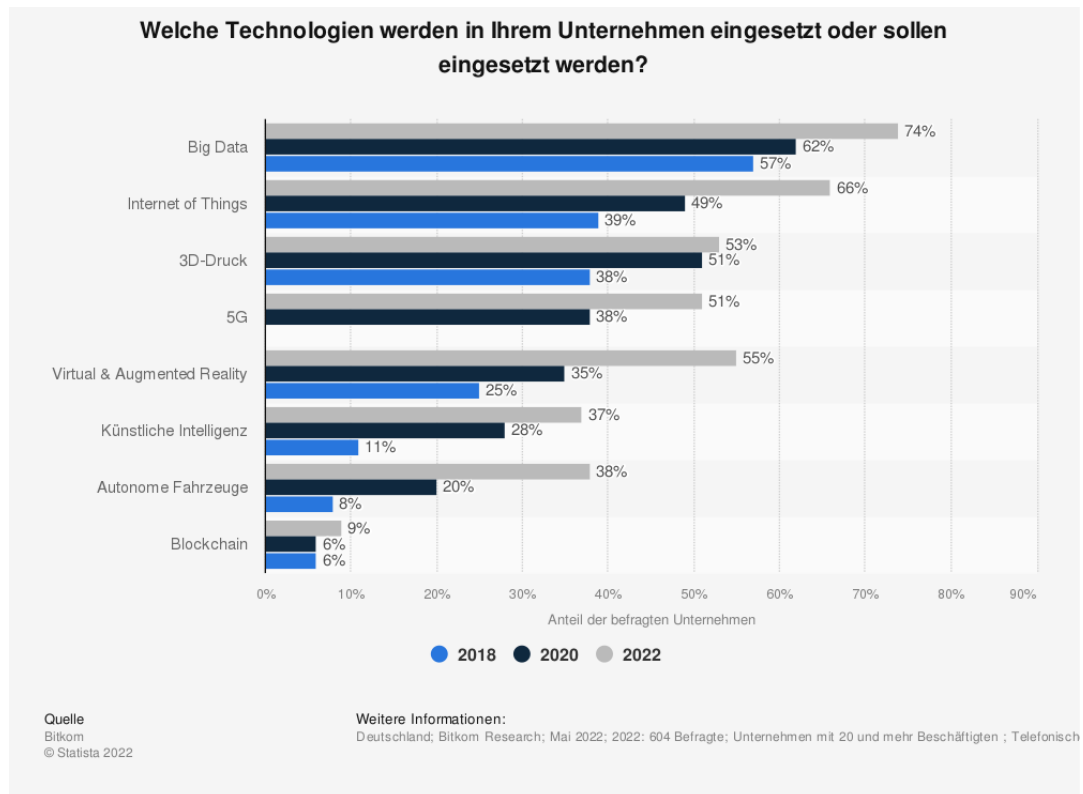


Abbildung 2.3: Eingesetzte Technologien im Jahr 2022  
Quelle: Statista 2022

Die unter Abbildung 2.4 sichtbare Statistik aus dem Jahr 2023 zeigt, dass die weltweit generierte Menge an Daten im Jahr 2012 ungefähr 6,5 Zettabyte betrug. Im Jahr 2020 hat sich die Menge auf 64,2 Zettabyte fast verzehnfacht. Im Jahr 2025 wird ein Anstieg auf bis zu 181 Zettabyte prognostiziert, die Tendenz bleibt wahrscheinlich steigend (Statista 2023).

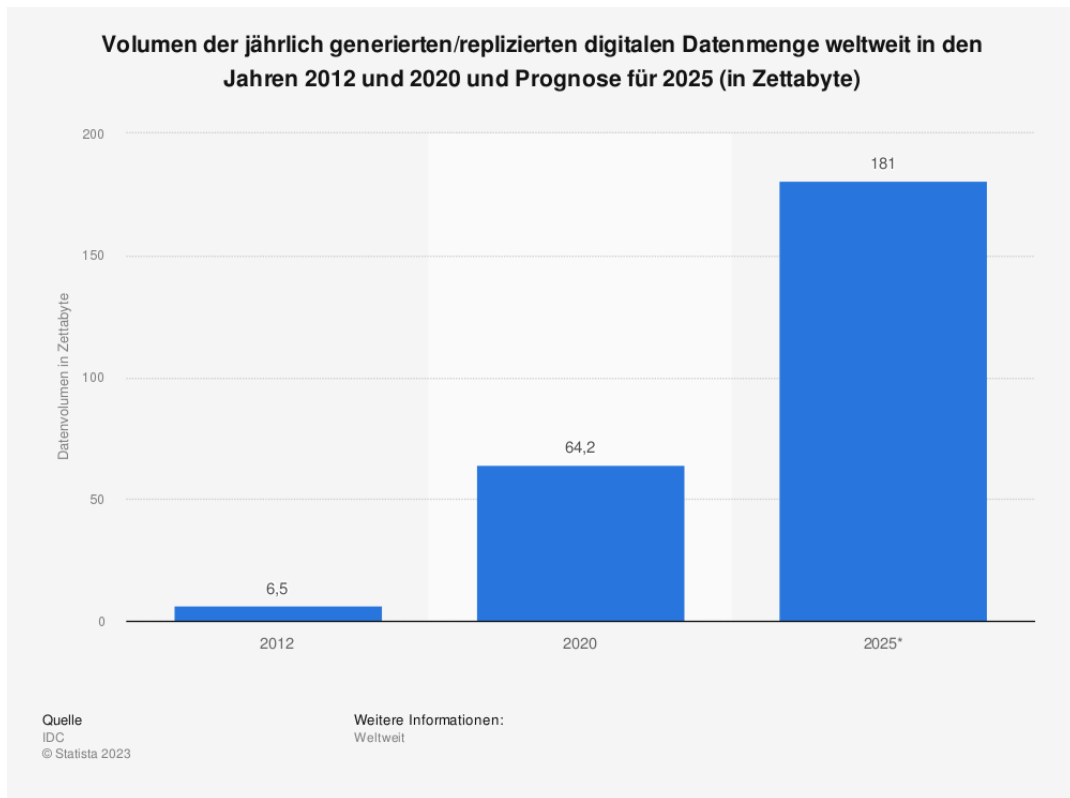


Abbildung 2.4: Prognose weltweit generiertes Datenvolumen bis 2025  
Quelle: Statista 2023

Die Definitionen von Big-Data variieren sehr stark, grundlegend bezieht sich der Begriff meistens auf die Speicherung, Analyse und Verwaltung von Datenmengen, die so groß und komplex sind, dass spezielle IT-Strukturen geschaffen werden müssen, um sie zu verarbeiten. State of the art ist dabei die Verteilung der Daten auf Cluster (verbundene Rechner), die sich die Arbeit untereinander aufteilen.[16], [17]

Standardmäßig wird Big-Data meistens nach den 3 V's: Volume, Velocity und Variety klassifiziert.[18] Später wurden dieser Charakterisierung mit Veracity und Value zwei weitere V's hinzugefügt.

- *Volume* - beschreibt die große Menge an Daten, die in Unternehmen anfallen und verwaltet werden müssen. Volume sagt aber nichts über deren Wert oder Zustand aus. Durch das Ausstatten von Geräten mit Sensorik und deren Vernetzung untereinander fallen immer mehr Daten, aus unterschiedlichen Quellen an. Wie in Abbildung 2.4 ersichtlich, steigt dieses Datenaufkommen kontinuierlich.[17]

- *Velocity* - beschreibt die Geschwindigkeit, mit der neue Daten generiert, ausgetauscht und verarbeitet werden müssen. Viele Unternehmen benötigen Anwendungen, die in der Lage sind, Daten in Echtzeit auszuwerten.[17], [16]
- *Variety* - entsteht durch die hohe Vielfalt an unterschiedlichen, Geräten, die unterschiedliche Daten erzeugen. Diese Daten können strukturiert oder unstrukturiert sein und verschiedene Datentypen und Formate beinhalten. Die Varianz der Daten hängt vom erzeugenden Medium ab und reicht von XML Dateien über rohen Text bis zu Bildern und Videos.[17], [16]
- *Veracity* - beschreibt die Unsicherheit, die durch unterschiedliche Datenquellen entsteht. Die geforderte Datenqualität ist nicht immer gegeben und muss oft durch Transformationen hergestellt werden. [16]
- *Value* bezieht sich, darauf, welchen Nutzen der Anwender aus den gesammelten Daten ziehen kann. Aus diesem Grund ist es wichtig, die Relevanz der gesammelten Daten zu bewerten, um eine effiziente Datenarchitektur zu schaffen, die Mehrwerte für ein Unternehmen erzeugen kann.[16]

## 2.3 Methodische Vorgehensweise

In Abschnitt 2.2 wurde ein allgemeiner Überblick über Big-Data und dessen Charakterisierung gegeben. Für die Entwicklung einer prototypischen Big-Data Architektur und einem Handlungsleitfaden zur Implementierung, ist es wichtig, diese Definition in den richtigen Kontext zu setzen. Um die Relevanz von Big Data in der Produktion zu verstehen und die Anforderungen an Datenarchitekturen zu erheben, wurde eine Literaturrecherche durchgeführt. Die Ergebnisse der Recherche wurden anschließend mithilfe der „qualitativen Inhaltsanalyse“ untersucht. Aufbauend auf dieser Analyse wurden Kategorienbäume erstellt, die Aufschluss über die in der Produktion relevanten Daten und die Anforderungen an die Architekturen geben. Die Vorgehensweise und die erarbeiteten Kategorienbäume werden in den folgenden Abschnitten erläutert.

### 2.3.1 Qualitative Inhaltsanalyse

Die qualitative Inhaltsanalyse ist eine Methode zur Auswertung von Literatur, mit dem Ziel den Text zu verstehen und daraus relevante Informationen zu exzerpieren. Die erarbeiteten Informationen dienen der Beantwortung einer Forschungsfrage oder dem Finden neuer Thesen. Es handelt sich um eine

regelbasierte, systematische Vorgehensweise zur Textinterpretation nach dem folgenden Prozess: [19][20]:

- Textmaterial in Kontext zur Fragestellung setzen: Textmaterial suchen und Relevanz für die Fragestellung feststellen
- Systematische Vorgehensweise: Durchführen der qualitativen Inhaltsanalyse nach einem festgelegten Ablaufmodell
- Regelbasiertes Vorgehen: Die Auswahl und Bewertung von Texten geschieht anhand von festgelegten Regeln
- Kategorisierung: Die Rechercheergebnisse werden in Kategorien eingeteilt
- Güte feststellen: Bewerten der Rechercheergebnisse in Bezug auf Transparenz, Objektivität und Reproduzierbarkeit

Nach Mayring gibt es drei praktikable Grundformen der Interpretation, die je nach Problemstellung besser geeignet sein können.[20, S.67]:

- Zusammenfassung - der Text soll abstrahiert und auf die wesentlichen Inhalte reduziert werden
- Explikation - Erweiterung des Textverständnisses durch Hinzufügen zusätzlicher Materials, um den Sachverhalt genauer zu beschreiben
- Strukturierung - Einordnen des Materials durch Bildung eines Querschnitts, um bestimmte Themen aussortieren zu können

Aufgrund der großen Menge an Referenzliteratur im Big-Data Umfeld wurde die Interpretationsform der Zusammenfassung gewählt.

### **2.3.2 Interpretative Vorgehensweise**

Das Kernelement der qualitativen Inhaltsanalyse ist das Einordnen der Exzerpte in Kategorien. Findet diese Kategorisierung durch Zusammenfassung, direkt am Textmaterial statt, so spricht man oft von „induktiver Kategorienbildung“.[19, S.72] Konkret bedeutet kategorisieren, dass relevante Teile des Textes extrahiert und durch Überbegriffe beschrieben werden. Unwichtige Inhalte werden gestrichen. [20] Zuerst müssen die Forschungsfragen geklärt werden, zu denen recherchiert wird. In Bezug auf diese Masterarbeit soll die Inhaltsanalyse Antworten auf die folgenden Fragen liefern:

1. Welche Bedeutung hat Big Data in der Produktion?
2. Welche, in der Produktion anfallenden, Daten sind relevant?
3. Welche Anforderungen an Big Data Architekturen stellt die Industrie/-Produktion?
4. Welche Herausforderungen entstehen bei der Implementierung?

Im nächsten Schritt müssen **Selektionskriterien** und ein **Abstraktionsgrad** festgelegt werden. Das Selektionskriterium gibt Aufschluss darüber, ob ein Text für die Beantwortung der Forschungsfragen geeignet ist. Die Selektionskriterien für diese Arbeit sind zwei Fragen, die in Bezug auf den Text mit Ja oder Nein beantwortet werden müssen, wobei Ja bedeutet, dass die Quelle für die Beantwortung der Fragestellung geeignet ist.

1. Wird im Text auf die Bedürfnisse an Big-Data Anwendungen von Produktionsbetrieben eingegangen?
2. Werden im Text Anforderungen an die Funktionalität von Big-Data Architekturen genannt?

Wenn diese Fragen mit Ja beantwortet werden können, wird das Ergebnis entweder einer Kategorie zugeordnet oder es wird bei Bedarf eine neue Kategorie eingeführt. Um sicherzustellen, dass die Kategorien nicht aus dem Kontext fallen, wurde ein **Abstraktionsgrad** eingeführt. Der angewandte **Abstraktionsgrad** bezieht sich nicht auf allgemein gültige Themen im Big-Data Umfeld, sondern auf einen spezifischen Anwendungsfall:

*„Die aus der Analyse erarbeiteten Kategorien müssen im Umfeld einer Produktion anwendbar sein können.“*

Trifft die Aussage, die im Abstraktionsgrad getroffen wurde, zu, so ist die Quelle, im Kontext der Forschungsfrage relevant und kann zur weiteren Analyse verwendet werden. Wenn nicht, wird die Quelle nicht weiter bearbeitet.

Als Hilfe für die Durchführung der Interpretation wurde ein eigens angefertigtes Strukturblatt verwendet, welches unter Anhang A [5.3] ersichtlich ist. Dieses Strukturblatt enthält die Selektionskriterien, den Abstraktionsgrad, die Forschungsfragen und allgemeine Daten zur Quelle wie Titel, Autor, Verlag und Datum. Die Vorgehensweise entspricht im Wesentlichen dem Ablaufmodell der induktiven Analyse nach Mayring [20, S.86]. Dieses Ablaufmodell ist in Abbildung 2.5 dargestellt.

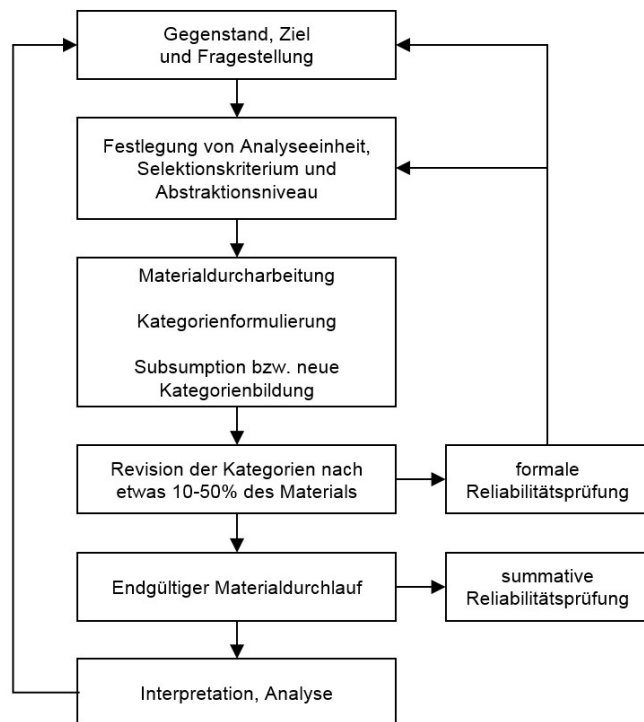


Abbildung 2.5: Ablaufmodell induktive Kategorienbildung  
Quelle: [20]

### 2.3.3 Recherche Ergebnisse

Im folgenden Abschnitt wird auf die Ergebnisse der Literaturrecherche und ihre Bewertung durch die qualitative Inhaltsanalyse eingegangen. Zu diesem Zweck wird zuerst kurz das System der Kategorienbäume erläutert. Danach werden die ausgearbeiteten Kategorienbäume der vier Kernfragen vorgestellt und die gewonnenen Erkenntnisse erläutert.

Abschließend werden die Ergebnisse zusammengefasst.

### Kategorienbaum

Die Inhaltsanalyse nach Mayring liefert Kategorienbäume, die aus den Rechercheergebnissen abgeleitet werden können. Abbildung 2.6 zeigt die grundlegende Struktur eines solchen Kategorienbaums. Aus der Recherche ergibt sich ein Kategoriensystem bestehend aus einer Hauptkategorie und wenn notwendig bis zu n-Subkategorien.



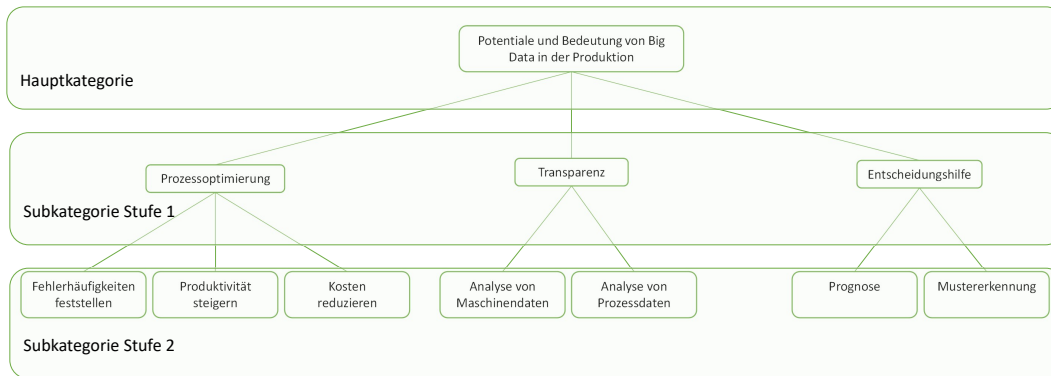


Abbildung 2.6: Kategorienbaum Struktur Übersicht  
Quelle: eigene Darstellung

## Relevante Daten

Die Kategorie „relevante Daten“ gibt Aufschluss darüber, welche Daten aus der laufenden Produktion wertbringend verarbeitet werden können.

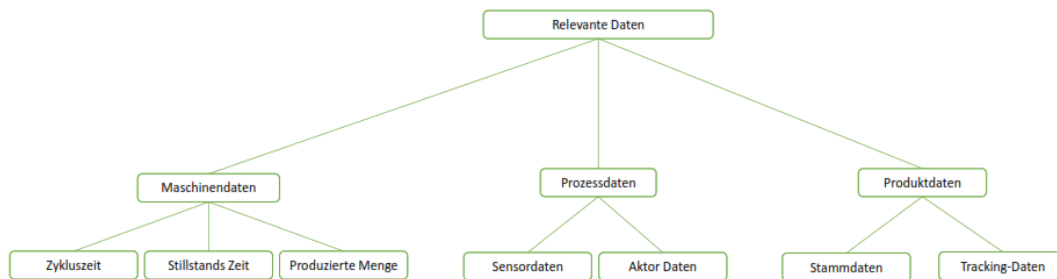


Abbildung 2.7: Kategorienbaum Relevante Daten  
Quelle: eigene Darstellung

*Maschinendaten* werden von Maschinen und Anlagen zur Verfügung gestellt. Maschinendaten sind produktionsrelevante Informationen, die im Betrieb entstehen. Beispiele für wichtige Maschinendaten sind: Produzierte Menge, Energieverbrauch, Zykluszeit und Stillstandszeit. Maschinendaten sind jene Daten, die für die Bewertung der Produktivität eine große Rolle spielen, dazu mehr im Abschnitt 2.3.3.[21], [22]

*Prozessdaten*, sind artverwandt mit Maschinendaten. Sie entstehen beim Betrieb einer Maschine, werden kontinuierlich gespeichert und enthalten Prozessparameter. Charakteristisch für Prozessdaten ist, dass es sich um physikalisch messbare Größen handelt. Sie stammen meistens von Sensoren oder

Aktoren. Beispiele für Prozessdaten sind: Drehmomente, Leistungsaufnahme, Kraft, Temperatur oder Druck. Prozessdaten werden mit Zeitstempel erfasst und ermöglichen eine Zeitreihenanalyse über den Zustand einer Maschine.[23] Prozessdaten spielen in der Umsetzung des Proof Of Concept in Kapitel 3.4 eine große Rolle.

Des Weiteren sind auch *Produktdaten* relevant. Produktdaten sind produktbezogene Informationen aus dem Produktionsprozess. Sie liefern Informationen über Produktparameter wie die gefertigte Menge eines bestimmten Produktes, Gewichte, Dimensionen und Stammdaten wie Artikelnummern und Stücklisten. Durch Sammeln von Produktdaten wird ebenfalls die Transparenz gefördert, da Produkte Chargen zugeordnet und so getrackt werden können.[21], [22]

## Potentiale und Bedeutung von Big Data in der Produktion

Die Subkategorie Prozessoptimierung behandelt primär die Optimierung von Produktionsprozessen, vorwiegend an automatischen Anlagen. Dabei spielt aus betrieblicher Sicht primär das Feststellen von Fehlerhäufigkeit, die Kostenreduktion und die Steigerung der Produktivität eine bedeutende Rolle. Optimierung wird durch die Analyse von Daten aus Sensoren, Mikrocontrollern und Steuerungen erreicht und soll Transparenz über die Produktionsprozesse gewährleisten.

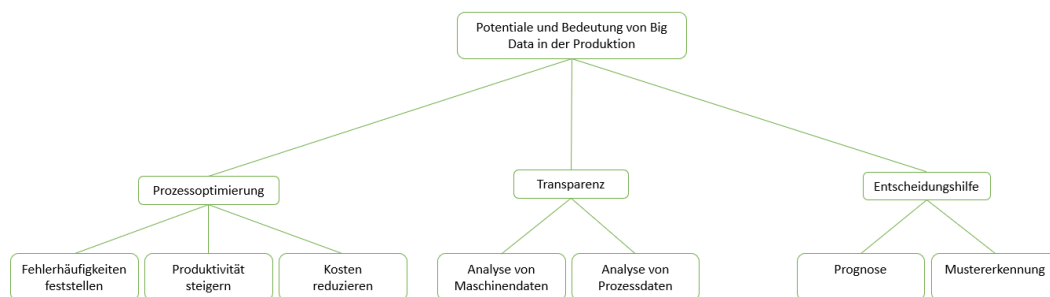


Abbildung 2.8: Kategorienbaum Potentiale und Bedeutung von Big Data in der Produktion

Quelle: eigene Darstellung

Für die Bestimmung der *Fehlerhäufigkeit* an Anlagenteilen können die Daten von Sensoren Aufschluss geben. Das Sammeln und Auswerten dieser Sensordaten soll langfristig ermöglichen, Ausfälle prognostizieren zu können. Im Umfeld von Big Data und Industrie 4.0 werden diese Methoden oft unter dem Begriff

„Predictive Maintenance“ zusammengefasst. Wichtig sind vorrangig physikalische Aufzeichnungen wie: Elektrische Leistungsaufnahme, Temperatur, Drehmomente oder Druck. Auch das Endprodukt soll auf häufige Fehler geprüft werden können. Dazu müssen die verbauten Komponenten nachverfolgbar gemacht werden. Dies geschieht zum Beispiel durch optische Identifikatoren wie Data-Matrix Codes oder mithilfe von RFID-Transpondern. Liegt ein fehlerhaftes Produkt vor, wird durch Nachverfolgbarkeit im Reklamationsfall zusätzliche Transparenz geschaffen.[24] Welche Technologien dafür eingesetzt werden, ist abhängig vom Produktionsprozess und der Geometrie des Produktes.

Für die *Steigerung der Produktivität* sind neben den oben genannten Prozessdaten von Sensoren vor allem „Maschinendaten“ interessant. Die Kennzahl „OEE“ (Overall Equipment Efficiency) gibt die real produktive Produktionszeit in Prozent an. Für die Berechnung der Kennzahl werden unter anderem, Maschinendaten wie: Stillstandszeit, produzierte Stückzahl und produzierter Ausschuss herangezogen.[25] In Kombination mit den erhobenen Prozessdaten können Bottlenecks und Schwachstellen von Maschinen und Anlagen identifiziert und optimiert werden.

*Kostenreduktion* spielt in jedem Unternehmen, immer, eine wichtige Rolle. Eine Möglichkeit der Kostenreduktion ergibt sich automatisch aus der erfolgreichen Umsetzung der zuvor genannten Methoden. Werden Schwachstellen an Anlagen durch die Analyse von Maschinen und Prozessdaten erkannt, kann durch entsprechende Gegenmaßnahmen die Produktivität gesteigert werden. Dadurch entstehen Einsparpotenziale. Zum Beispiel durch niedrigere oder sogar entfallende Rüst- und Stillstandszeiten. Die Erfassung und Verarbeitung von Maschinen und Prozessdaten geschieht mit dem Ziel *Transparenz* über den Produktionsprozess zu schaffen.[23]

Aus technischer Sicht bedeutet *Transparenz* die Möglichkeit in Echtzeit Prozessdaten und Maschinendaten abzugreifen, also den Status der Maschine verstehen zu können. *Transparenz* und durchgängiges Engineering sind miteinander verwandt, da die Datenmodelle des digitalen Zwillings, *Transparenz* über Prozesse und Zustände schaffen.

Gemäß dem fünften „V“ für „Value“ soll Big-Data in einem Unternehmen einen Mehrwert schaffen. Mehrwert kann unter anderem dadurch erreicht werden, dass mithilfe von Echtzeit Daten eine *Entscheidungshilfe* geschaffen wird. *Entscheidungshilfen* können Visualisierungen darstellen, die Menschen unterstützen sollen, indem komplexe Sachverhalte grafisch dargestellt werden. *Entschei-*

*dungshilfen* können aber auch automatisiert werden. Durch die Anwendung von Machine Learning Algorithmen wie der Klassifikation von Daten anhand ihrer Merkmale oder Regression zur Vorhersage zukünftigen Verhaltens, können Muster in Prozessen erkannt und Prognosen erstellt werden.

## Anforderungen an Big Data Architekturen

In produzierenden Unternehmen ist im Jahr 2023 die Rechenleistung und die verfügbare Speicherkapazität meistens nicht die Hauptanforderung an eine Big-Data Architektur. Die Hauptschwierigkeiten liegen darin, schwankende Datenmengen in Echtzeit verarbeiten zu können und gleichzeitig die Flexibilität des Systems aufrechtzuerhalten. Dieses Kategoriensystems zeigt die relevantesten Ergebnisse, die aus der Literaturrecherche ausgearbeitet wurden. An dieser Stelle muss darauf hingewiesen werden, dass IT-Security oft als Anforderung an Big-Data Architekturen genannt wird. In dieser Arbeit werden diese Themen nicht behandelt.

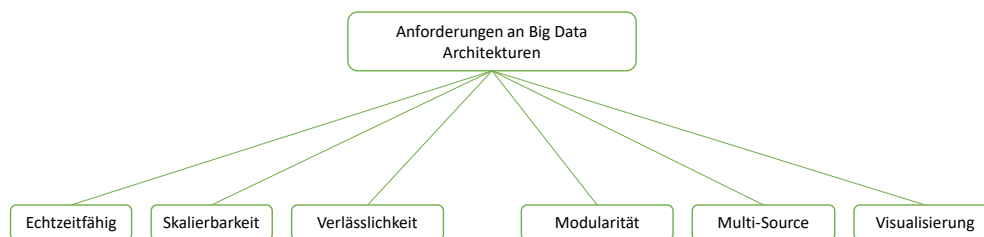


Abbildung 2.9: Kategorienbaum Anforderungen an Big Data Architekturen  
Quelle: eigene Darstellung

Ein System, das *echtzeitfähig* ist, gibt dem Anwender die Möglichkeit ad-hoc auf Veränderungen zu reagieren. Zum Beispiel auf die Veränderung von Prozessdaten wie dem Drehmoment, das von einem Sensor aufgezeichnet wurde und von einem CPPS mit einer Big-Data Anwendung geteilt wurde. Das bedeutet auch, dass eine gewisse Netzwerkbandbreite vorausgesetzt werden muss. *Echtzeitfähige* Architekturen haben die Fähigkeit Daten in hoher Geschwindigkeit zu sammeln, zu analysieren und verfügbar zu machen.[26]

*Skalierbarkeit* heißt, dass die Big-Data Anwendung auf schwankende Datenmengen reagieren kann. Wird etwa die Anzahl der angeschlossenen Anlagen von drei auf zwölf erhöht, steigt automatisch die zu verarbeitende Datenmenge. Erhöht sich also die Datenmenge muss auch das System „hoch-skaliert“ werden. Skalierbarkeit muss bei der Entwicklung einer Architektur von Anfang an

berücksichtigt werden [27, S.10]. Nach Strengtholt [28] bedeutet Skalierbarkeit in Bezug auf eine Architektur, dass Daten, Konsumenten und Nutzer flexibel miteinander verbunden sind und einander Einsicht in ihre Prozesse gewähren [28].

*Verlässlichkeit* bedeutet im Wesentlichen, dass die Anwendung die vom User geforderte Funktion erfüllt, Fehler des Users bis zu einem gewissen Grad abfangen kann, für den Anwendungsfall eine ausreichende Performance zur Verfügung stellt und sie gegen nicht autorisierte Zugriffe geschützt ist.[27, S.6]

Wie der Begriff Big-Data aussagt, handelt es sich um große Datenmengen, die gespeichert werden müssen. Die *Speicherkapazität* ist somit eine Kernanforderung jeder Big-Data Architektur. Eine standardisierte Richtgröße für die Speicherkapazität gibt es nicht, allerdings gilt auch hier das Credo der möglichen Skalierbarkeit, um adäquat auf Veränderungen reagieren zu können.

*Modularität* bedeutet im Zusammenhang mit Big-Data, aus Hardwaresicht, dass neue CPS mit wenig Aufwand, in das System, eingebunden werden können. Aus Softwaresicht sollte es möglich sein, unterschiedliche Applikationen zu verwenden, je nachdem was für den Anwendungsfall benötigt wird.[26] Beispielsweise soll sowohl Batch als auch Stream-Processing möglich sein. Unterschiedliche Datenbanktypen wie SQL, NoSQL und Time Series DB sollen unterstützt werden, genauso wie unterschiedliche Storage Technologien, wie Data Lakes, Data Warehouses und Blob Storage. Darüber hinaus werden unterschiedliche Analyse Tools und API's benötigt, die unterschiedliche Programmiersprachen unterstützen (Python, Scala, R, SQL etc.).

Die Big-Data Architektur muss gewährleisten, dass Daten aus verschiedenen Quellen verarbeitet werden können. Dies wird durch den Begriff *Multi-source* beschrieben. Die Herausforderung liegt darin, dass mehrere Quellen, unterschiedliche Datentypen, in unstrukturierter oder strukturierter Form bereitstellen. Diese Daten müssen vor oder nach dem Speichern bereinigt und gegebenenfalls transformiert werden, um analysiert werden zu können. Oft haben Unternehmen keinen Einfluss darauf, in welchem Format Maschinenhersteller Prozess und Maschinendaten zur Verfügung stellen.[27, S.27] Bei der Entwicklung einer Architektur muss im Vorhinein darauf geachtet werden, dass unterschiedliche Formate wie JSON, XML, CSV etc. verarbeitet werden können. Die Datenqualität muss ebenfalls sichergestellt werden, in manchen Fällen sind dafür separate Transformationsprozesse notwendig.

*Visualisierung* wird von Analysten benötigt, um sich zum Beispiel schnell ein Bild einer Fehlfunktion machen zu können. *Visualisierung* spielt aber vor allem im Reporting zur Unternehmensführung eine wichtige Rolle, um komplexe Sachverhalte verständlich präsentieren zu können. Die schnelle Verfügbarkeit von möglichst echtzeitnahen *Visualisierungen* spielen auch in der Steuerung und der Kapazitätsplanung eine wichtige Rolle.[26]

## Herausforderungen in der Implementierung

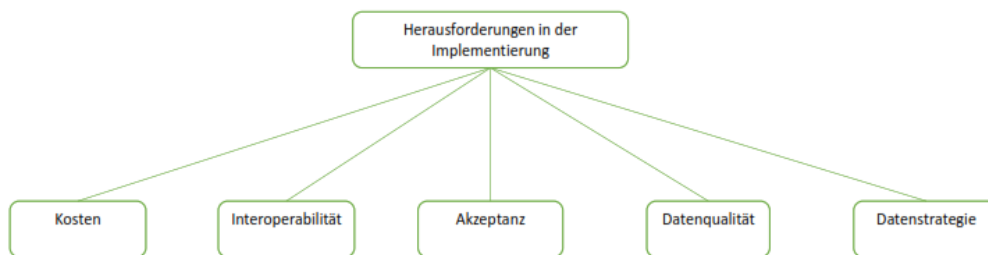


Abbildung 2.10: Kategorienbaum Herausforderung in der Implementierung  
Quelle: eigene Darstellung

Jede Implementierung einer neuen Technologie ist mit Veränderungen, Risiken und unvorhersehbaren Ereignissen verbunden. Aus dieser Recherche wurden diejenigen Punkte, die am meisten Risiken mit sich bringen, herausgearbeitet.

*Kosten* spielen in jedem Unternehmen eine Rolle, vor allem wenn der Mehrwert nicht kurzfristig erkennbar ist. Kosten entstehen durch die Anschaffung von Software, Hardware und das Upgraden bestehender Anlagen mit IOT tauglichen Komponenten. Kosten entstehen aber auch dadurch, dass fachkompetente Mitarbeiter benötigt werden.[29]

Unter *Interoperabilität* versteht man die Fähigkeit eines Systems, mit anderen Systemen zu kooperieren, ohne dass der User eingreifen muss und ohne dass alle Funktionen und Merkmale des jeweiligen anderen Systems bekannt sind. [28] Die Syntax der ausgetauschten Informationen spielt dabei eine wichtige Rolle, Standardformate wie CSV, XML oder JSON müssen definiert werden, um einen reibungslosen Prozess zu gewährleisten. Eine wichtige Rolle spielt auch die Datenübertragung, die nach standardisierten Protokollen wie MQTT, TCP/IP geschehen kann.

Wie in Abschnitt 2.3.3 erwähnt, muss es möglich sein, Daten aus verschiedenen Quellen verarbeiten zu können. Das bedeutet, dass die Daten auf *Datenqualität* wie Duplikate, Plausibilität und fehlende Einträge kontrolliert werden müssen, um die Ergebnisse nicht zu verfälschen.[29] Die benötigte Genauigkeit muss bei der Entwicklung und beim Anbinden neuer Systeme getestet und definiert werden.

Die *Akzeptanz* und das Verständnis von Big-Data und dessen Vorteile muss als Erstes bei Entscheidungsträgern in einem Unternehmen erzeugt werden. Begegnet man einem Big-Data Projekt nicht mit Akzeptanz und Offenheit, können durch Scheitern desselben viele Ressourcen und finanzielle Mittel verloren gehen.[29] Eine langfristige Folge davon könnte sein, dass der technologische Anschluss an den Stand der Technik und somit die Wettbewerbsfähigkeit verloren geht.

Eine *Datenstrategie* ist notwendig, um zu definieren, welche Ziele durch den Einsatz von Big-Data Technologien erreicht werden sollen. Durch den Anstieg von Datenmengen steigt auch die Komplexität. Die Prozesse müssen stetig auf neue Daten und Technologien zugeschnitten werden, um zu gewährleisten, dass die richtigen Informationen zur richtigen Zeit zur Verfügung gestellt werden können. Dazu gehört die Erkenntnis, welche Daten als relevant zu betrachten sind.[30, S.15-16]

#### **2.3.4 Zusammenfassung der Ergebnisse**

Aus der Literaturanalyse lassen sich einige Kerneigenschaften ableiten, die für die Integration einer Big-Data Architektur in einer Produktion wichtig sind. Im Kern sind dies die Möglichkeit, Daten aus dem laufenden Betrieb in Echtzeit zu erfassen und in einer skalierbaren, funktional modular erweiterbaren Architektur so weiter verarbeiten zu können, dass ein Mehrwert für das Unternehmen entsteht. Mehrwert bedeutet in diesem Zusammenhang die Möglichkeit, Fehler und Ausfälle an Anlagenteilen früh genug erkennen und prognostizieren zu können, dadurch die Produktivität zu erhöhen und Kosten einzusparen. Des Weiteren soll Transparenz über die Produktionsprozesse geschaffen werden. Der Status der Systeme soll so echtzeitnah wie möglich visuell abrufbar sein. Somit ist das Hauptziel dieser Arbeit eine in der Produktion bei der Zumtobel Group möglichst universell einsetzbare, skalierbare Daten-Architektur zu schaffen.

# 3 Proof-Of-Concept

Nach der Definition von Big-Data treten große Datenmengen, in hoher Geschwindigkeit und hoher Varianz in einem Unternehmen auf und benötigen spezielle Strukturen um verarbeitet zu werden. In diesem Kapitel wird die Umsetzung eines Prototyps für eine Big-Data Architektur, geeignet für die Nutzung in einer Produktionsumgebung, besprochen. Der Prototyp soll die in Kapitel 2 herausgearbeiteten Anforderungen und Funktionen erfüllen und als Proof-Of-Concept (POC) dienen.

## 3.1 Vorgehensweise

Die methodische Vorgehensweise, die für die Umsetzung des POC angewandt wurde, stützt sich auf die Publikation „A Design Science Research Methodology for Information Systems Research“, kurz DSRM, aus dem Jahr 2008[31]. Die Vorgehensweise baut auf folgenden Aktivitäten auf[31]:

1. Problem identifizieren und Motivation beschreiben
2. Ziele einer entwickelten Lösung beschreiben
3. Entwurf und Entwicklung eines Artefaktes
4. Demonstration
5. Evaluation
6. Kommunikation

Den Kern dieser Methode bildet das „Artefakt“. Das Artefakt ist ein speziell für das Forschungsproblem entwickeltes Objekt, das eine Lösung, für ein Forschungsproblem bietet. Es gibt keine Vorschrift, die besagt, dass die angegebene Reihenfolge exakt eingehalten werden muss. Dies richtet sich nach dem für das Thema passenden Forschungsansatz, der problemorientiert, zielorientiert, entwicklungsorientiert oder kundenorientiert sein kann. Es ist somit möglich beim zielorientierten Ansatz bereits mit dem zweiten Schritt zu starten.[31] Dieser Prozess ist in Abbildung 3.1 visuell dargestellt.



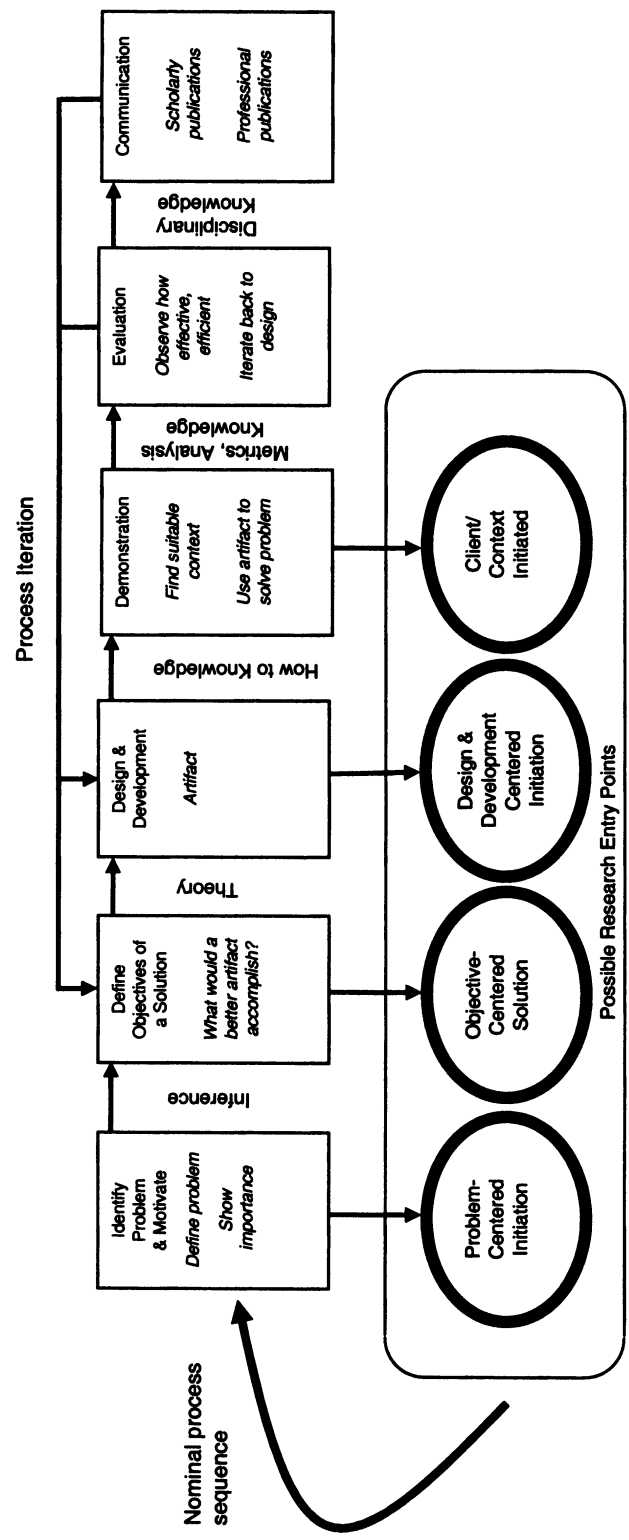


Abbildung 3.1: DSRM Prozess  
Quelle:[31]

Der zielorientierte Ansatz wird oft angewandt, wenn Wünsche aus der Wirtschaft oder Forschung durch die Entwicklung eines Artefaktes erfüllt werden können.[31] Da für diese Arbeit das Ziel definiert war, eine skalierbare und universell anwendbare Big-Data Architektur zu entwickeln, wurde der zielorientierte Ansatz gewählt.

## **Problem Identifizieren**

Die Datenverarbeitung, konkreter die Erfassung und Analyse von Daten aus dem laufenden Betrieb wird in Produktionsunternehmen oft vernachlässigt, obwohl darin großes Potenzial für die Erhaltung der Wettbewerbsfähigkeit, Kostenreduktion und Optimierung liegt. Oft ist unklar, welche Daten zur Auswertung relevant sind. Das Erheben, Speichern und die Verarbeitung von großen Datenmengen benötigt spezielle Methoden und Werkzeuge. Ein System das diese Methoden und Werkzeuge zur Verfügung stellt, wird im Kontext dieser Masterarbeit als Big-Data-Architektur bezeichnet. Für die Entwicklung einer solchen Architektur ist es notwendig, die zu verarbeitenden Daten zu verstehen und die Anforderungen des Unternehmens an die Architektur zu kennen.

## **Ziele**

Die Arbeit hat im wesentlichen drei Kernziele:

1. die Definition der Anforderungen, welche die Produktion an eine Big-Data-Architektur stellt
2. die Entwicklung einer möglichst universell einsetzbaren Big-Data-Architektur für die Anwendung in der Produktion
3. die Entwicklung eines Handlungsleitfadens, der als Orientierungshilfe für die zukünftige Umsetzung von Big-Data-Projekten in der Produktion innerhalb der Zumtobel Group dienen soll

## **Konzeption und Entwicklung**

Um die Anforderungen an die Datenarchitektur zu ermitteln, wurde eine qualitative Inhaltsanalyse anhand einer gewissen Auswahl an Literatur durchgeführt. Aus den Ergebnissen konnten allgemeine Anforderungen, Kernfunktionen, Potenziale und Herausforderungen abgeleitet werden, die für die Implementierung und den Betrieb einer Big-Data Architektur zutreffen. Die Big-Data Architektur als Artefakt ist so aufgebaut, dass es die grundlegenden Ziele Skalierbarkeit

und universelle Anwendbarkeit unter Berücksichtigung der definierten Anforderungen erreichen kann. Getestet wurde im Umfeld der Produktion, an realen Maschinen, mit deren realen Daten.

## **Demonstration**

Die Anforderungen an die Big-Data-Architektur wurden anhand einer Literaturrecherche mit anschließender qualitativen Inhaltsanalyse definiert und bewertet. Die Funktionen der Big-Data Architektur wurden anhand einer prototypischen Implementation demonstriert. Der konkrete Anwendungsfall ist vom Partnerunternehmen bereits im Vorfeld definiert worden. Für die Demonstration wurden Daten aus der Prägestation der Tecton Tragschienenanlage verwendet. Diese Daten wurden in der entwickelten Big-Data-Architektur gesammelt, ausgewertet und dargestellt. Abgeleitet aus den Rechercheergebnissen und den Erfahrungen während der Implementierung wurde ein Handlungsleitfaden erstellt. Dieser Handlungsleitfaden ist als Flussdiagramm in Kapitel 4 abgebildet und in Textform näher erklärt.

## **Evaluation**

Das Projekt wurde evaluiert durch Vergleich des neu entwickelten Prototyps mit der vorhandenen Architektur, welche in Abschnitt 3.2 beschrieben ist. Die Evaluation fand primär auf funktionaler Ebene statt und stellt sich die grundlegenden Fragen, ob im Vergleich zum Ursprungsprototypen eine Verbesserung festgestellt werden kann und ob die Architektur zukunftssicher ist. Es wurde geprüft, ob der Prototyp, die in Abschnitt 2.3 herausgearbeiteten Anforderungen erfüllt. Eine weitere Analyse des Handlungsleitfadens ist zum Zeitpunkt der Abgabe dieser Masterarbeit nicht möglich. Für dessen genauere Bewertung ist ein Nachfolgeprojekt notwendig, bei dem nach der vorgestellten Methodik gearbeitet wird. Bisher fand eine Beurteilung durch Kollegen und Beteiligte am Projekt statt. Der Handlungsleitfaden ist stark spezifiziert auf den Anwendungsfall innerhalb der Zumtobel Group und hat keinen Anspruch auf Allgemeingültigkeit. In Abschnitt 4 ist der Handlungsleitfaden näher beschrieben und abgebildet.

## **Kommunikation**

Die Ergebnisse der Masterarbeit wurden im Unternehmen in mehreren Präsentationen geteilt und demonstriert. Während des Projektverlaufes fanden regelmäßige Meetings statt, mit dem Zweck die Beteiligten und Sponsoren über den aktuellen Projektstatus zu informieren. Im Kreis der Teilnehmer befanden sich sowohl Mitarbeiter aus dem Bereich Industrial und Manufacturing Engineering

als auch aus der Werksleitung. Die methodische Vorgehensweise, wie auf diese Masterarbeit angewandt, ist unter Abbildung 3.2 grafisch dargestellt.

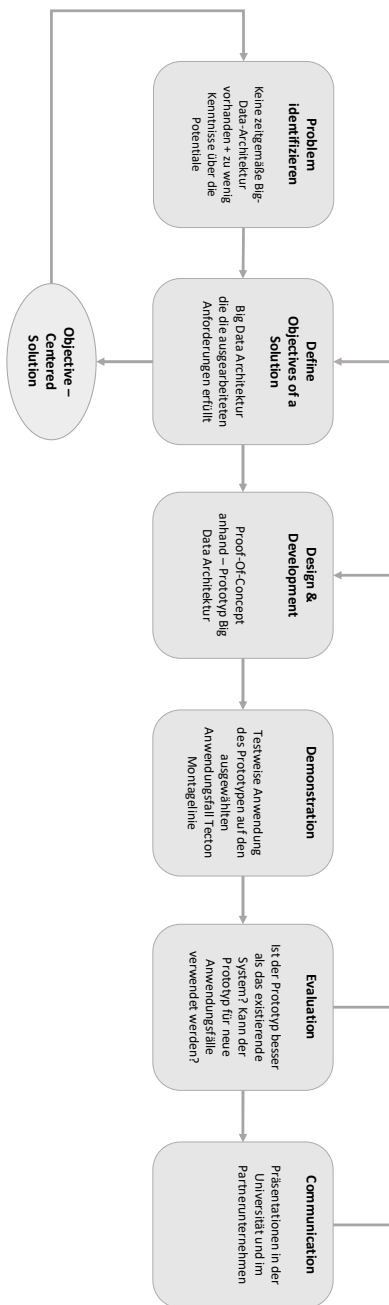


Abbildung 3.2: DSRM angewandt  
Quelle: eigene Darstellung

## 3.2 Ausgangssituation

Zum Startzeitpunkt dieser Masterarbeit gab es in der Produktion der Zumtobel Group, am Standort Dornbirn, eine einzige Anwendung, die als Vorstufe einer Big-Data Anwendung gesehen werden konnte. Dabei handelt es sich um einen Testaufbau an der Montagelinie des Produktes Tecton. In dieser Montagelinie befindet sich eine Prägestation deren akkurate Funktion, kritisch für die Qualität des Produktes ist. Da es in der Vergangenheit bereits Probleme mit der Station gab, wurde hier ein Versuch für einen datengetriebenen Überwachungsprozess gestartet. Im folgenden Abschnitt wird kurz das Produkt Tecton erklärt, was zum besseren Verständnis gegenüber dem Prozess dienen soll. Anschließend wird der Fertigungsprozess an der Montagelinie oberflächlich erklärt und zum Schluss wird auf die Architektur des bestehenden Testaufbaus eingegangen.

### 3.2.1 Tecton Produktbeschreibung

Tecton ist ein modulares Lichtbandsystem und gemessen in Stückzahlen eines der wichtigsten Produkte der Zumtobel Group. Als Lichtbänder werden Leuchten-Systeme bezeichnet, die es ermöglichen mehrere Leuchtenmodule linear hintereinander zu montieren und zu betreiben. Das Licht bildet somit ein Band. Die Anwendungen sind vielseitig, finden sich aber meistens in Produktions- und Lagerhallen, aber auch im Retail Bereich zum Beispiel in Supermärkten wieder. Neben den vielfältigen Einsatzfeldern ist die Modularität von Tecton einer der Hauptvorteile des Produktes, theoretisch kann das Lichtbandsystem mehrere hundert Meter lang werden. Herzstück des Portfolios ist die Tecton Tragschiene. Bei der Tragschiene handelt es sich um ein rollgeformtes Stahlblechprofil, in das eine Stromschiene eingeschoben wird. Die Stromschiene besteht aus mehreren Kunststoffteilen aus Polypropylen, die auf die gewünschte Länge miteinander verbunden werden. In die Stromschiene werden Kupferdrähte eingerollt, anschließend werden die Blechschiene und die Stromschiene miteinander verheiratet. Zusätzlich wird ein Verbinder montiert, der es ermöglicht mehrere Tragschienen auf die gewünschte Länge miteinander zu kontaktieren. Die Stromschiene hat mehrere Funktionen. Als Erstes erfüllt sie die Montagefunktion, sie kann abgependelt oder fest an die Decke montiert werden. Als Zweites dient sie der Spannungsversorgung der daran befestigten Leuchten. Fast jede Leuchte der Zumtobel Group kann mit dem richtigen Adapter an das Tragschienen-system angebunden werden.

Zum besseren Verständnis ist das Produkt in Abbildung 3.3 im Schnitt ersichtlich. Die einzelnen Bestandteile sind beschriftet. Die Abbildung stammt aus den CAD-Daten, die von der Zumtobel Group zur Verfügung gestellt wurden.

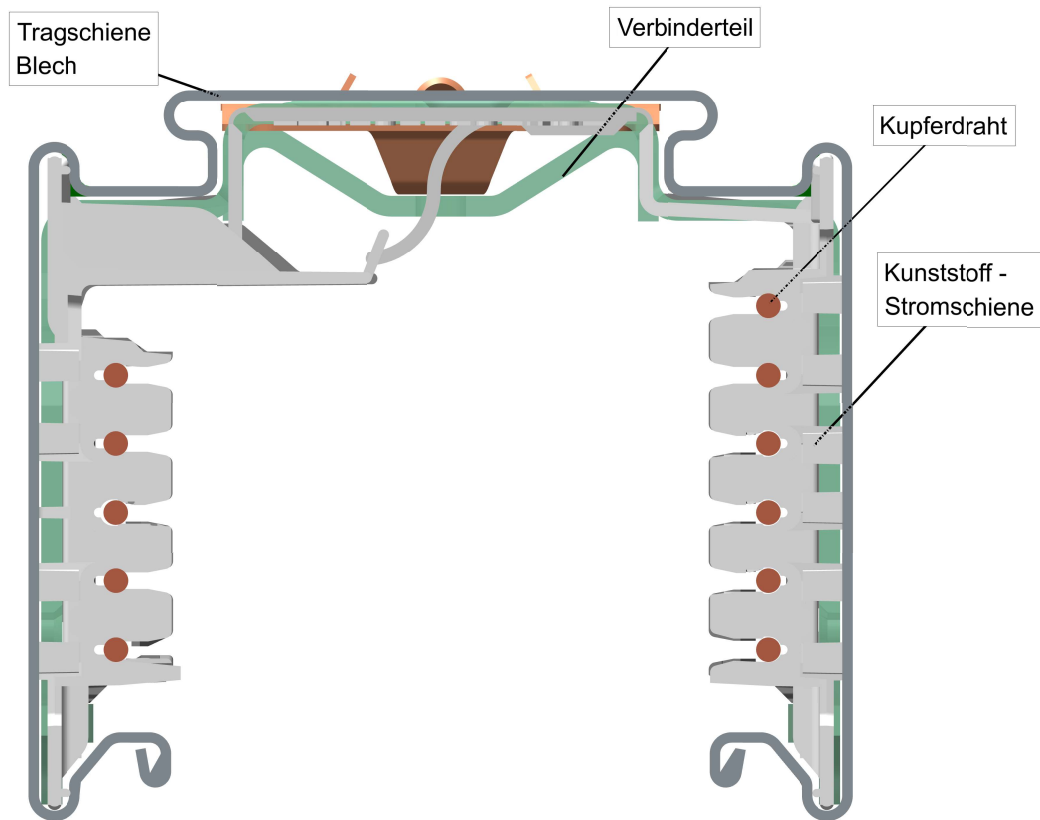


Abbildung 3.3: Querschnitt Tecton Tragschiene  
Quelle:Zumtobel Group

### 3.2.2 Tecton Montageprozess

Bei der Tecton Tragschienen Montagelinie handelt es sich um eine verkettete Anlage. Verkettet bedeutet in diesem Fall, dass die Montagelinie und die Rollformlinie hintereinander geschaltet sind. Eine Übersicht der Stationen, nach Produktionsschritten findet sich nachfolgend. Unter Abbildung 3.4 ist das Anlagenlayout dargestellt. Danach folgt eine kurze Beschreibung des Fertigungsprozesses.

1. Station: Draht einziehen
2. Station: Draht einpressen

3. Station: Verheiraten der Profile
4. **Station: Prägen**
5. Station: Verpackung
6. Station: Palettieren
7. Separate Anlage: Schiene profilieren und zuführen

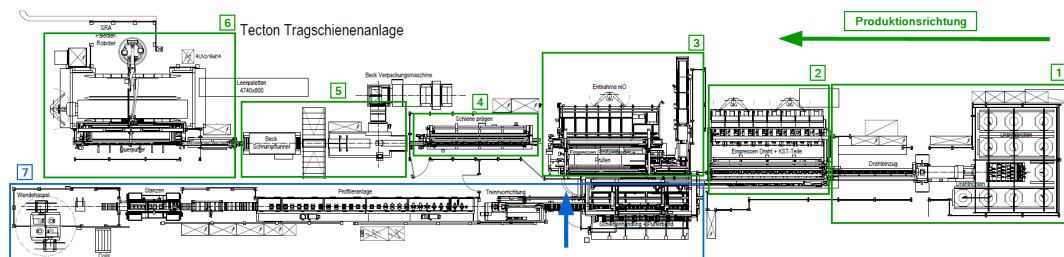


Abbildung 3.4: Tecton Anlagenlayout  
Quelle: Zumtobel Group

In Station 1 wird der Kupferdraht aus Behältern gezogen und gerichtet. In Station 2 werden die Stromschienelemente eingelegt und verbunden, der Draht wird über die verbundene Stromschiene gezogen, heruntergedrückt und mit Rollen in die vorgesehenen Rillen eingerollt. Blau eingerahmt ist die Rollformlinie ersichtlich, hier werden die Blechprofile gefertigt und in einen Pufferspeicher gefahren. Der blaue Pfeil zeigt die Richtung an aus der die Profile den Pufferturm in Richtung Montagelinie verlassen. In Station 3 wird das Tragschienelementprofil aus dem Pufferturm in die Montagelinie übernommen und die Stromschiene in das Tragschienelementprofil eingezogen. Anschließend wird das Verbinderteil montiert. Die Tragschiene wird zu Station 4, der Prägestation weitertransportiert. In dieser Station findet der Prägeprozess statt, mit dem die Stromschiene in der Tragschiene befestigt wird. Dieser Prozess ist kritisch, da die Haltekraft der Prägung maßgeblich für die Funktion und die Qualität des Endproduktes ist. Der Prägevorgang wird mit je zwei Stempeln, die links und rechts des Profils geführt sind durchgeführt. Nach dem Prägen wird das Produkt in Station 5 in Folie verpackt und in Station 6 von einem Roboter auf eine Palette abgestapelt. Abbildung 3.5 zeigt die im Produkt angebrachte Prägung vergrößert anhand der CAD-Daten.

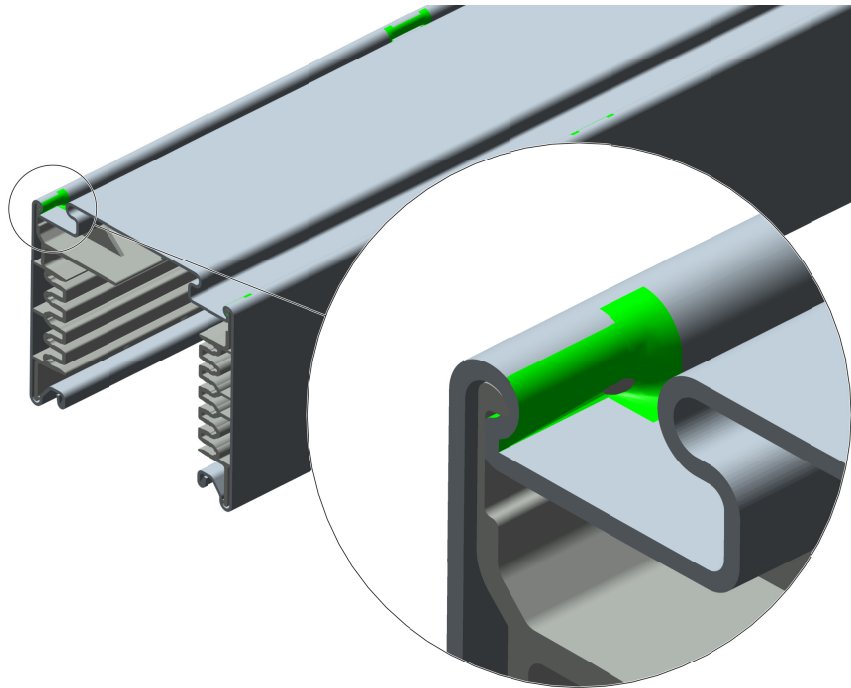


Abbildung 3.5: Prägung TECTON Tragschiene  
Quelle:Zumtobel Group

In der Industrialisierungsphase des Produktes wurde festgestellt, dass die Stempel schnell verschleifen und dadurch keine ausreichende Haltekraft mehr gegeben ist, das Produkt ist somit Ausschuss. Daraus entstand die Idee, Sensoren an der Prägestation anzubringen, die den Kraftaufwand und den zurückgelegten Weg der Stempel aufnehmen. Diese Sensoren sind über Ethernet mit einem Industrie-PC (IPC) verbunden, auf dem die Messwerte in einer lokalen Datenbank gespeichert und ausgewertet werden. Fallen die Messwerte Kraft oder Weg unter einen bestimmten Schwellwert, wird ein Signal an die Steuerung der Anlage gesendet und der bearbeitete Teil als Nicht-In-Ordnung ausgeschleust. Die genauen Auswertelgorithmen sind nicht Teil dieser Arbeit und werden deshalb nicht näher beschrieben. Die historischen Daten werden anschließend in eine SQL Datenbank gespeichert und als CSV-Datei zur Visualisierung in Power BI zur Verfügung gestellt. Auch wenn diese Anwendung im Produktionsbetrieb funktioniert, hat sie doch einige entscheidende Nachteile:

- Nicht Echtzeitfähig
- Keine Skalierbarkeit gegeben
- Begrenzte Analysemöglichkeiten der erhobenen Daten



- Die Visualisierung in Power BI ist nur beschränkten User-Gruppen zugänglich

Abbildung 3.6 zeigt die Architektur, die in der Ausgangssituation beschrieben wurde

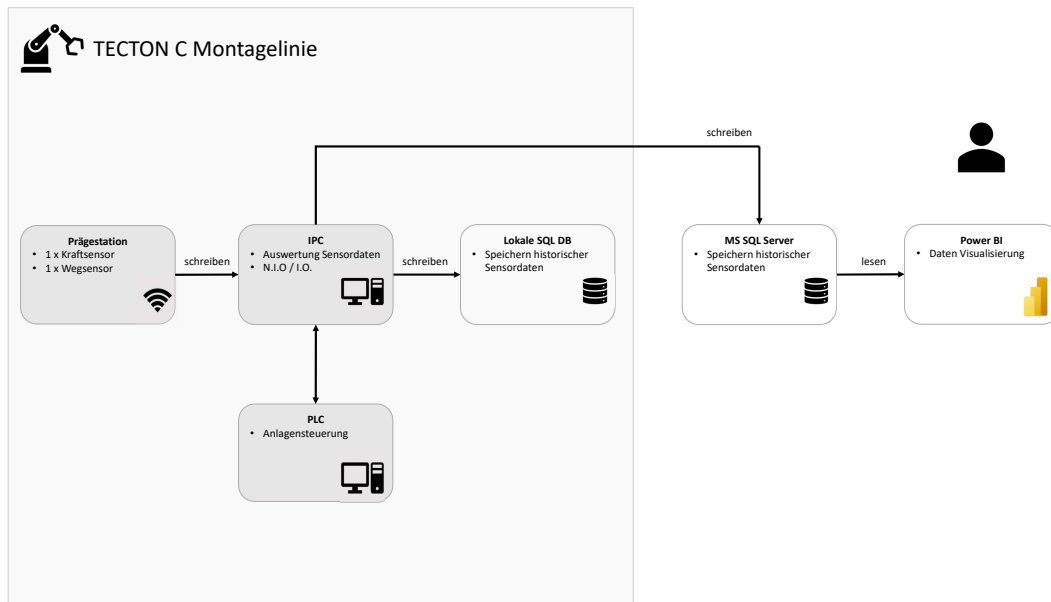


Abbildung 3.6: Architektur Ausgangssituation  
Quelle: eigene Darstellung

### 3.3 Die Referenzarchitektur

Die Umsetzung des Prototyps basiert auf den Grundprinzipien und Ideen die von Pietheine Strengholt in „Data Management At Scale“ [1] vorgestellt werden. Neben Skalierbarkeit steht Modularität und Flexibilität im Vordergrund dieser Architektur. Um diese Eigenschaften erfüllen zu können, spielt das „Domain-Driven-Design“ Prinzip in der Implementierung eine große Rolle. Dieser Ansatz verfolgt das Ziel, eine Applikation zu entwickeln, die zum betrachteten Prozess oder Problem passt und dessen Anforderungen am besten erfüllt. Die von Strengholt vorgeschlagene skalierbare Big-Data-Architektur besteht oberflächlich betrachtet aus den drei Hauptkomponenten Data Providers, Data Layer und Data Consumers. Diese Architektur wurde als Referenzarchitektur für die Implementierung des Prototyps herangezogen und wird im folgenden Abschnitt genauer erklärt. Unter Abbildung 3.7 ist die Architektur grafisch dargestellt.

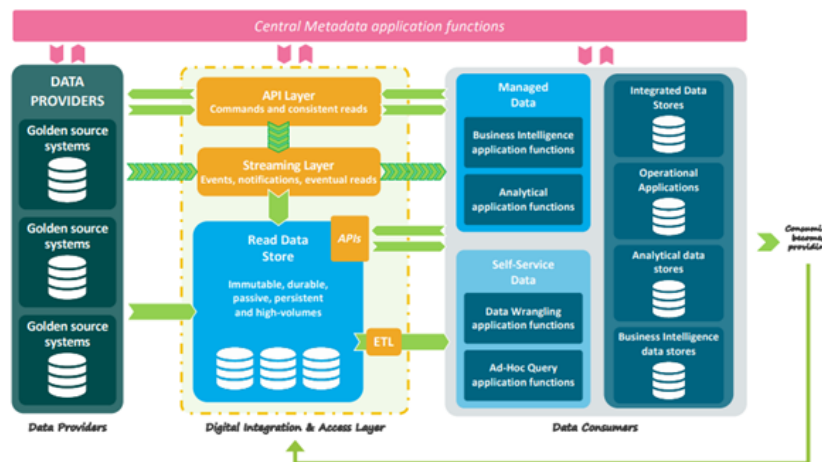


Abbildung 3.7: Referenzarchitektur Data Management At Scale  
 Quelle:[1, S.183]

### 3.3.1 Data Providers

Data Providers bestehen aus den sogenannten Golden Source Systems. Golden Sources sind Anwendungen oder Applikationen, die authentischen Daten zur Verfügung stellen. Golden Datasets sind wiederum die Daten, die von den Golden Sources zur Verfügung gestellt werden, diese Datensätze müssen akkurat und vollständig sein. Golden Datasets müssen nicht aus der ursprünglich erzeugenden Quelle stammen.[28]

Data Providers können, unter anderem, Cyber-Physikalische-Systeme, Datenbanken oder Data Lakes sein.

### 3.3.2 Data Layer

Oberflächlich betrachtet, ist der Data Layer das Herzstück der Architektur und verbindet die Data Providers mit den Data Consumers. Die Integration des Data Layer stellt sicher, dass alle Daten durch dieselben Schnittstellen laufen. Das Konzept des Data Layers soll zusätzliche Transparenz darüber schaffen, wie Applikationen miteinander verbunden sind und welche Routen Daten nehmen. Immer wenn Daten zwischen Anwendungen verschoben werden, sind Transformationen notwendig. Dieser Prozess wird mit ETL (Extract, Transform, Load) abgekürzt, in manchen Fällen ist die Reihenfolge ELT (Extract, Load, Transform) passender. Egal, ob die Daten in Batches oder als Stream übertragen werden, eine Integration ist immer notwendig. Strengholt bezeichnet dies als das „Data Integration Dilemma“ [1, S.19]. Betrachtet man den Data Layer, eine Ebene tiefer, so besteht er aus mehreren, standardisierten, Sub-Architekturen

die unterschiedliche Technologien verwenden und unterschiedliche Anwendungsfälle abdecken. Diese drei Sub-Architekturen sind die Read-Only Data Store Architecture (RDS), die API Architecture und die Streaming Architecture. Diese Struktur macht das Handling von verschiedenen Datenmodellen und Formaten und somit die Komplexität von Big Data möglich und stellt sicher, dass die Architektur flexibel bleibt[1, S.48]. Wichtig ist zu erwähnen, dass auch hier der Domain-Driven Approach gilt, es müssen nicht alle drei Optionen benutzt werden, sondern jene, die für die Daten und die Anwendungen am besten passen.

### 3.3.3 Read-Only Datastores

Die Read-Only Datastores Architektur, im folgenden RDS genannt, macht schreibgeschützte Daten für die Wiederverwendung zugänglich. Datenbereinigung, Verarbeitung und ETL-Prozesse sind durch ihren Einsatz umsetzbar genauso wie das Speichern von historischen Daten [1, S.47]. RDS sind ähnlich wie Data Warehouses, im direkten Vergleich gibt es aber technische Unterschiede. So sind RDS spezialisiert auf Datenanalyse Prozesse, können große Volumen speichern, unterschiedliche Schemas verarbeiten und haben die Fähigkeit, die Datenqualität zu kontrollieren. RDS bestehen oft aus Daten aus stark frequentierten Quellen, die nicht verfälscht oder verloren gehen dürfen. Daher bilden RDS, Kopien dieser Daten, die vom Consumer nur gelesen, aber nicht überschrieben werden können. Das bedeutet, dass nach dem RDS-Prinzip eine Lese-Datenbank und eine Schreib-Datenbank existieren. Dieses System birgt den Vorteil, dass die beiden Datenbanken separat voneinander skaliert werden können. Die separate Skalierbarkeit ist deshalb interessant, weil Schreibprozesse mehr Performance benötigen als Leseprozesse. Dieses Prinzip wird „Command and Query Responsibility Segregation“, kurz CQRS genannt. Nach diesem System wird für jede Operation zwischen Command oder Query unterschieden. Durch diese Trennung ist es möglich, die Performance der jeweiligen Anwendung zu verbessern. Durch die zentrale Stellung im Data Layer wird immer, wenn Provider und Consumer Daten austauschen ein RDS erstellt. RDS-Systeme müssen gewisse Grundeigenschaften erfüllen, um zu funktionieren, dazu gehören:

- Unveränderlichkeit, RDS dürfen nur gelesen aber nicht beschrieben werden
- Nur Golden Datasets dürfen geladen werden
- RDS sind eng verknüpft mit deren Quell-Systemen und müssen daher speziell für diese entworfen werden (Domain Driven Design). Sie folgen deren Kontext und Datenmodellen.

Typische Aufgaben von RDS-Systemen sind das Sicherstellen der Datenqualität und die Aufbereitung der Daten für die weitere Nutzung durch Consumer. Ein Beispiel für RDS sind Delta Tables, die von Azure Databricks zur Verfügung gestellt werden.

### 3.3.4 API Architecture

Unterschiedliche Datenmodelle, Formate und Anwendungen haben unterschiedliche Anforderungen, die von unterschiedlichen Diensten geleistet werden müssen. Für die Verteilung der Aufgaben auf diese Dienste und Anwendungen ist die API-Architektur zuständig. Die API-Architektur erlaubt es unterschiedliche Methoden und Programmiersprachen zur Entwicklung von Daten-Applikationen zu verwenden. Die API-Architektur lässt sich effizient mit RDS und Event-Streams verbinden, dabei übernehmen API's die Lese und Schreibprozesse der über RDS oder Streams zur Verfügung gestellten Datasets. Die API-Architektur stellt eine Oberfläche dar, die Consumern und Producern unterschiedliche Services zur Verfügung stellt. Diese Services können genutzt werden, um die Transformation und Kommunikation von Daten zu realisieren. Im Verbund mit RDS können API's eingesetzt werden um die CQRS-Prinzipien zu erfüllen. Da für jeden Anwendungsfall eine spezifische API verwendet werden kann, ist das API-Konzept ein Kernbestandteil des Domain-Driven-Design Prinzips und ermöglicht einen modularen Aufbau innerhalb des Data Layers.[28]

Zum Beispiel wird für die Implementierung des Prototyps in dieser Masterarbeit die Databricks-Spark-Scala API verwendet, um das Event-Streaming von MQTT-Nachrichten zu realisieren.

### 3.3.5 Streaming Architecture

Die Streaming Architektur wird zur Echtzeit-Bereitstellung von Ereignissen und Nachrichten verwendet. Streaming wird vor allem dann eingesetzt, wenn hohe Datenvolumen verarbeitet werden müssen. Stream Processing reagiert auf das Auftreten von Events in Echtzeit und steht im harten Gegensatz zum Batch-Processing, bei dem manuell zugeführte Datasets verarbeitet werden. Streaming-Kommunikationsmuster werden oft als asynchron bezeichnet, da der Client auf eingehende Nachrichten wartet. Das empfangende System muss nicht sofort auf den Eingang einer Nachricht reagieren und das sendende System kann trotzdem weiterarbeiten, ohne auf eine Antwort warten zu müssen. Nach Ablauf einer gewissen Zeit, nach dem Nachrichteneingang wird die Verbindung oft gekappt und die Nachrichten gelöscht. Streaming-Plattformen arbeiten meistens „Event Driven“. Das bedeutet, dass Informationen über auftretende Änderungen ausgetauscht werden. Eine solche Änderung wird Event genannt. Änderun-

gen sind meistens Zustandsänderungen. Ein plastisches Beispiel wäre die Messung eines Temperatursensors, der eine Temperaturänderung registriert. Die Temperaturänderung ist eine Zustandsänderung, also ein Event, das der Sensor per Stream weiterleitet. Beispiele für Streaming-Plattformen sind Kafka, RabbitMQ oder ActiveMQ.[28, S.121-123]

### 3.3.6 Event Driven Architecture

Streams tauschen Informationen über auftretende Änderungen aus, diese Änderungen werden Event genannt. Tritt eine Änderung eines Zustandes auf, wird eine Nachricht, an eine Streaming-Plattform gesendet. Als Nachrichten-Plattform fungieren sogenannte Broker. Die Architektur besteht aus den drei Hauptbestandteilen Event Producer, Broker und Event Consumer. Der Broker ist zuständig für die Verwaltung der eingehenden Nachrichten. Nachrichten von Producern treffen beim Broker ein und werden von diesem an die jeweiligen Consumer weitergeleitet. Der Broker ist in der Lage Nachrichten von mehreren Producern gleichzeitig zu erhalten und diese an verschiedene Consumer weiterzuleiten. Außerdem bestimmt der Broker, ob Nachrichten nach dem Weiterleiten aufbewahrt oder gelöscht werden. Producer publishen ihre Nachrichten auf ein oder mehrere Topics und senden diese an den Broker, die Consumer subscriben auf ein gewünschtes Topic und erhalten die Daten vom Broker. Event Processing unterscheidet zwischen drei Typen:

- Simple Event Processing (SEP) - beim SEP werden Nachrichten über Veränderungen ohne weitere Bearbeitung weitergeleitet.
- Event Stream Processing (ESP) - beim ESP werden Operationen wie Filter, Berechnungen oder Queries auf den Datenstreams ausgeführt.
- Complex Event Processing (CEP) - Das CEP kann für die Mustererkennung in Datenströmen eingesetzt werden.

Streaming Anwendungen und API-Architekturen arbeiten eng zusammen, da APIs benötigt werden, um Producer und Consumer an das System anzubinden und um die Datenströme zu steuern. Die Quellen der Daten werden als Event Source bezeichnet und ihr Speicher oder ihre Zieldestination als Event-Sink. APIs werden aber nicht nur für diese Verknüpfung benötigt, sondern auch um die Daten vor oder nach der Ankunft am Zielort weiterzuverarbeiten.[1, S.136]

### 3.3.7 Event Sourcing

Das Management von Events kann auf unterschiedliche Weise erfolgen, aufgrund der hohen Relevanz für diese Arbeit wird das Event Sourcing hervorgehoben

und kurz erläutert. Event Sourcing bedeutet, dass alle Events in einem unveränderlichen Speicher im „append mode“ abgelegt werden, vergangene Events werden also nicht gelöscht und neue Events werden immer zu den historischen hinzugefügt. RDS eignen sich besonders gut für solche Speicher. Durch Event Sourcing lassen sich vergangene Events nachverfolgen und rekonstruieren, da alle Änderungen als eine Sequenz von Events gespeichert werden. Die Daten von Consumern werden in RDS abgelegt, von wo aus Producer oder API's Zugriff auf sie haben. Um zusätzliche Flexibilität und Modularität zu gewährleisten, werden RDS immer auf die jeweiligen Daten zugeschnitten. Gegeben durch die Eigenschaften von RDS sind historische Daten unveränderlich gespeichert und können als Kopien für die weitere Verarbeitung zur Verfügung gestellt werden.[1, S.137-139]

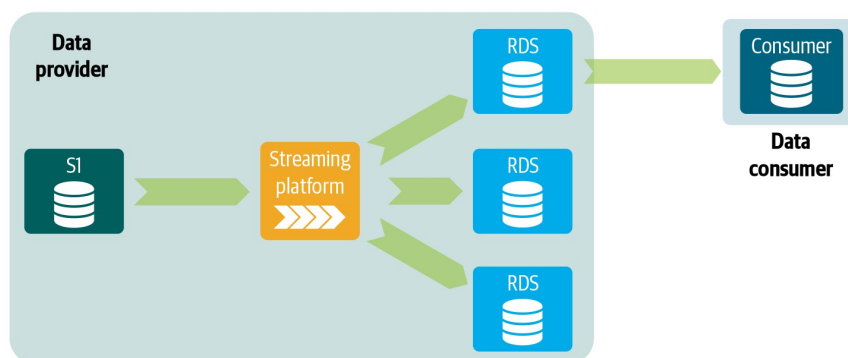


Abbildung 3.8: Event Sourcing Schema  
Quelle:[1, S.138]

## Stream Consumption

Für das Handling von Streams auf der Consumerseite gibt es ebenfalls eine hohe Anzahl an unterschiedlichen Varianten. Auch hier gilt das Domain-Driven Prinzip, der Consumer übernimmt die Transformation der Daten und gibt die Struktur der Consumption vor. Wie eingangs erwähnt gibt es viele verschiedene Möglichkeiten, die Consumer Prozesse umzusetzen, auf die in dieser Arbeit nicht eingegangen wird. Aufgrund der Relevanz für diese Arbeit wird im Folgenden genauer auf Event-Routing in der Cloud eingegangen.

### Event Routing in der Cloud

Bei dieser Methode werden Applikationen, die in einer Cloud gehostet sind, als Event Plattform verwendet. Beispiele dafür sind die Anwendungen Azure IOT-Hub und Azure Event Hub. Diese Applikationen erfüllen ähnliche Zwecke wie

Broker und haben den Vorteil, dass Events innerhalb der Cloud relativ einfach „geroutet“ werden können. So können etwa Streams über den Event-Hub direkt an Datenbanken oder Datalakes weitergeleitet werden.[1, S.143-145]

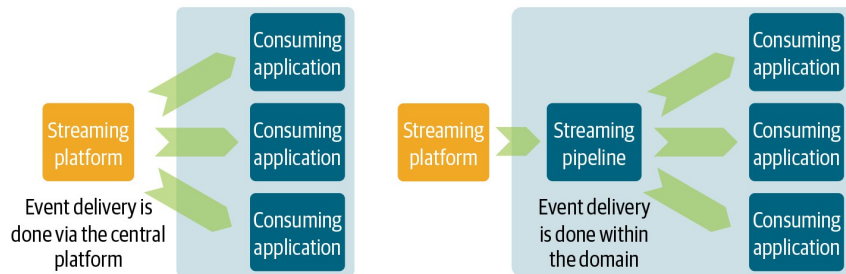


Abbildung 3.9: Event Routing in der Cloud  
Quelle:[1, S.145]

### 3.3.8 Data Consumers

Applikationen können Datenquellen (Golden Sources) oder Data Consumers, also Empfänger sein. Data Consumers speichern und integrieren die Daten und können sowohl intern als auch extern einer Organisation angesiedelt sein. Die in Abbildung 3.7 angeführten Speichermedien werden als Domain-Data-Stores (im Folgenden kurz als DDS) bezeichnet. Diese Speicher fungieren als Data Consumers und halten Daten, die in anderen Anwendungen produziert wurden. Domain Data Stores sind „Domain Driven“, das bedeutet, dass sie auf den jeweiligen Anwendungsfall und die jeweiligen Daten zugeschnitten sind und unterschiedliche Charakteristiken aufweisen können. DDS können sowohl Data Consumer als auch Data Producer sein. Dies geschieht, wenn durch Integration und Transformation von Daten in einem DDS ein neues „Golden Dataset“ entsteht, das mit anderen Applikationen geteilt wird. So wird ein DDS vom Consumer zum Producer, also einer „Golden Source“. Vergleichbar mit den Data Providers können auch Data Consumers unterschiedliche Formen annehmen. Data Consumers können zum Beispiel APIs, Visualisierungstools, Datenbanken oder Data Lakes sein.[1]

## 3.4 Umsetzung der Big-Data-Architektur

In diesem Kapitel wurde bisher auf die angewandte methodische Vorgehensweise eingegangen, danach wurde der Ausgangszustand erläutert und der Anwendungsfall der für die Umsetzung des Proof-Of-Concept dient, erklärt. Anschließend wurde die Referenzarchitektur vorgestellt, auf welcher die Umsetzung aufbaut. Damit wurde die notwendige Basis geschaffen, um mit der Konzeptionierung und Entwicklung des Prototyps zu beginnen. In diesem Abschnitt werden die Systemgrenzen der entwickelten Architektur gezogen, bevor umfassend auf die Umsetzung eingegangen wird, zum Schluss folgt eine Analyse der Ergebnisse.

### 3.4.1 Systemgrenzen

Bevor unter 3.4.2 detailliert auf die konkrete Umsetzung eingegangen wird, müssen die Systemgrenzen und Schnittstellen des Prototyps abstrahiert betrachtet werden. Grob können die Systeme in ein Hardware und ein Softwaresystem, mit ihren jeweiligen Subsystemen eingeteilt werden. Die Hardwareseite vertritt die Anlagensicht, während die Softwareseite die Big-Data Architektur betrachtet. Dabei werden die Funktionen der jeweiligen Bausteine übergeordnet untersucht bevor diese in die einzelnen Layer aufgebrochen und detailliert erklärt werden. Bei einer zukünftigen Erweiterung der Anwendung ist diese Systemdefinition hilfreich um eine erste Übersicht über die geforderten Funktionen zu erhalten. In Abbildung 3.10 ist das Gesamtsystem abstrahiert, grafisch dargestellt. Auf der Hardwareseite befindet sich die Anlage, mit ihrem Subsystem, der Prägestation. Eine Ebene tiefer kann das Subsystem Prägestation in ein weiteres Subsystem unterteilt werden, das Subsystem Sensoren. An dieser Stelle wird deutlich, dass jedes System eine beliebige Anzahl an Subsystemen beherbergen kann. Die Sensoren haben die Funktion, Daten zu erheben und zu teilen, genauso wie die Steuerung. Die Steuerung hat zusätzlich noch eine Auswertefunktion, da sie entscheiden kann ob ein Teil in Ordnung oder nicht in Ordnung ist. Das Softwaresystem, also die Cloud Anwendung besteht grob betrachtet aus den vier Subsystemen IOT/Event-Hub, API, Datenbank und Visualisierung. Der IOT/Event-Hub hat die Aufgabe Nachrichten vom Broker zu empfangen und diese zu verteilen, dabei übernimmt der IOT-Hub das Event Routing in der Cloud. Das Subsystem API ist dafür zuständig, die Daten aus den Hubs zu lesen, die notwendigen Transformationen, Berechnungen und Analysen durchzuführen und die Daten dann in das Subsystem Datenbank zu schreiben. Die Datenbank hat die Hauptaufgabe, die Daten zu lesen, zu schreiben und für andere Systeme bereitzustellen. Das Subsystem Visualisierung liest die Daten aus dem Quellsystem, in diesem Fall ist das die Datenbank, und stellt diese



dem Anwender zur Verfügung. Zwischen den beiden Systemen befindet sich der Broker als zentrale Schnittstelle, er hat die Aufgabe, Daten zu empfangen und zu verteilen.

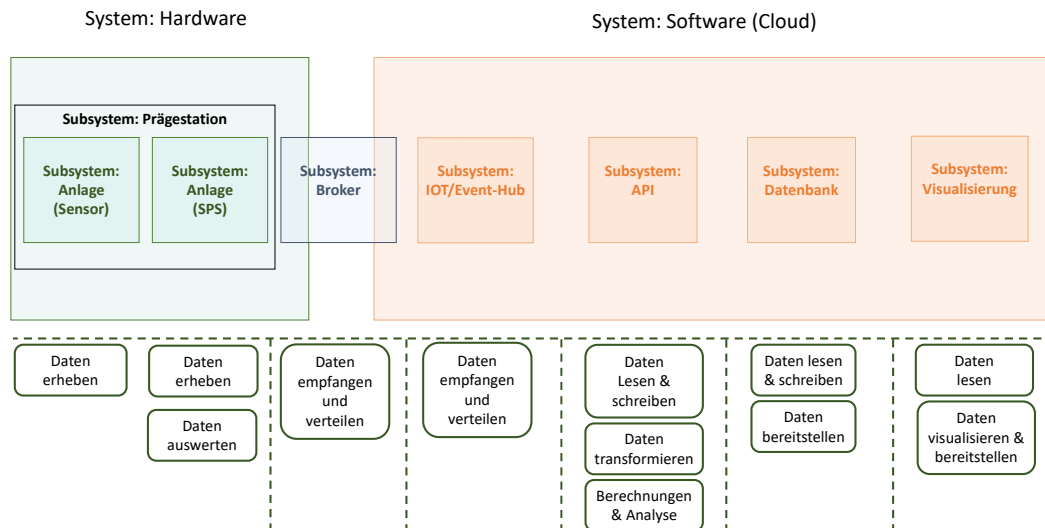


Abbildung 3.10: System Übersicht  
Quelle: Eigene Darstellung

### 3.4.2 Prototyp Architektur

Eine Grundanforderung des Partnerunternehmens an die Architektur war die Umsetzung innerhalb einer Cloud Anwendung. Da die Zumtobel Group vorrangig Microsoft Tools verwendet und bereits über eine Azure Subscription verfügt, wurde auf deren Wunsch hin die Azure Cloud verwendet. Der Data Layer und der Data Consumer Layer wurden komplett in Azure implementiert. Die Umsetzung des Proof-Of-Concept geschieht anhand eines Prototyps, der sich an der zuvor erläuterten Referenzarchitektur von Strengholt[28] orientiert. Abbildung 3.11 zeigt den Aufbau der Architektur und die verwendeten Bausteine der drei Layer Data Providers, Data Layer und Data consumers. Die Technologien, die von Azure in der Cloud bereitgestellt werden, werden Ressourcen genannt. Best Practice ist es alle Ressourcen für eine bestimmte Anwendung in einer Ressourcengruppe zu sammeln. Der folgende Abschnitt beschreibt die verwendeten Technologien und zeigt mögliche Alternativen auf.

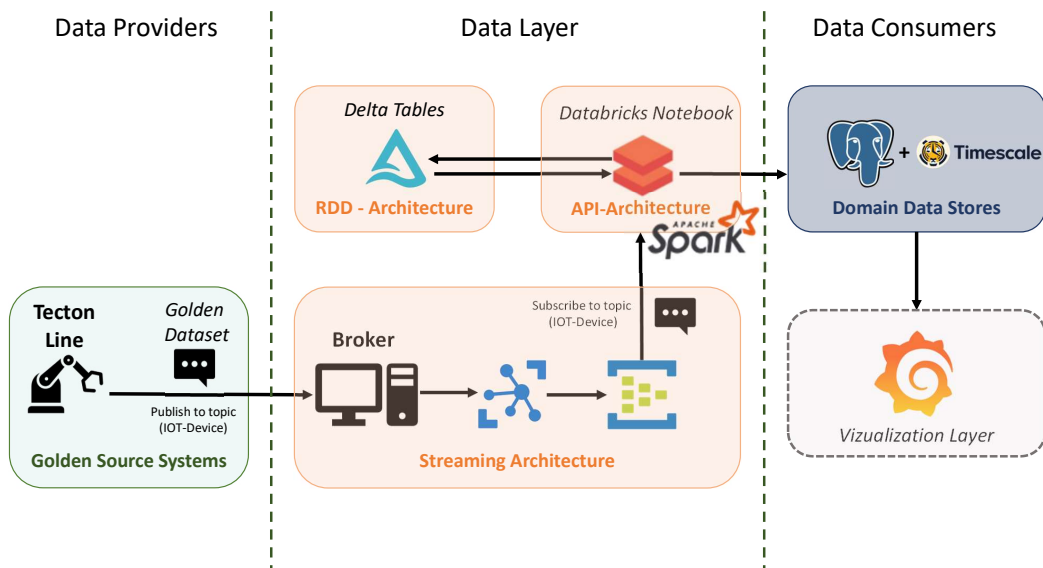


Abbildung 3.11: Prototyp Architektur  
Quelle: Eigene Darstellung

## Data Providers

Das Golden Source System des Prototyps ist die Steuerung der Tecton Montagelinie. Hier werden Prozessdaten und Maschinendaten erhoben. Die Prozessdaten sind in diesem Fall die Messwerte der Sensoren. Die Auswertung, ob ein Teil gut oder schlecht ist, geschieht direkt auf der Steuerung, damit im Fehlerfall der Schlecht-Teil sofort ausgeschleust werden kann. Die Messwerte der Sensoren werden von der Steuerung in JSON geparkt. Das Schema ist im nachfolgenden Listing ersichtlich.

```

1  "sensorid": "Werkzeug1_Seite2",
2  "praegid": "1",
3  "series": [{
4      "ts": 1629907927404,
5      "Weg": 13.530992,
6      "Kraft": 417
7  }
8  ]

```

Zusätzlich werden auch Maschinendaten erhoben, in diesem Fall handelt es sich dabei um Stillstandszeiten, Zykluszeiten, produzierte Teile und die Anzahl der schlechten Teile. Aus diesen Daten wird die Kennzahl „Overall-Equipment-Efficiency“, abgekürzt als „OEE“, berechnet. Bei der OEE handelt es sich um

eine Kennzahl, die Aufschluss über die Produktivität von Maschinen und Anlagen gibt.[32] Die Maschinendaten werden ebenfalls in JSON geparkt, das Schema ist im folgenden Listing ersichtlich.

```

1  "series": [{
2      "Produktionszeit": 7.7,
3      "Stillstandszeiten": 0.3,
4      "AnzahlStillStaendeGeplant": 1,
5      "AnzahlStillStaendeUngeplant": 5,
6      "Zykluszeit": 0.006944444,
7      "Stueckzahl": 898,
8      "Betriebszeit": 7.4,
9      "StueckGef": 879,
10     "StueckNIO": 19
11     }
12 ]

```

Die OEE ist das Produkt aus den Werten: Technische Verfügbarkeit, Leistungsgrad und Qualitätsgrad.

$$OEE = \textit{TechnischeVerfuegbarkeit} \cdot \textit{Leistungsgrad} \cdot \textit{Qualitaetsgrad} \quad (3.1a)$$

Die Technische Verfügbarkeit berechnet sich durch Division der effektiven Produktionszeit mit der geplanten Produktionszeit. Die effektive Produktionszeit ist dabei die geplante Produktionszeit abzüglich der Stillstandszeiten. Der Betrachtungszeitraum ist immer eine Schicht, also 7,7 Arbeitsstunden.

$$\textit{Technische Verfuegbarkeit} = \frac{\textit{Produktionszeit} - \textit{Stillstandszeit}}{\textit{Produktionszeit}} \quad (3.2a)$$

Der Leistungsgrad berechnet sich durch Multiplikation der gefertigten Stückzahl mit der Zykluszeit pro Teil, anschließend wird dieser Faktor durch die Effektive Produktionszeit dividiert. Die Zykluszeit ist jene Zeit, die benötigt wird um genau ein Stück eines Artikels zu produzieren. In anderen Worten bedeutet das, dass die Ist-Leistung durch die Soll-Leistung dividiert wird.[32]

$$\textit{Leistungsgrad} = \frac{\textit{Stueckreal} * \textit{Zykluszeit}}{\textit{EffektiveProduktionszeit}} \quad (3.3a)$$

Zum Schluss wird der Qualitätsgrad berechnet. Dazu wird zuerst die Differenz aus den gesamt gefertigten Teilen und den fehlerhaften Teilen gebildet und anschließend durch die Gesamtanzahl dividiert. In anderen Worten bedeutet das, dass die Anzahl der Gut-Teile durch die Anzahl der insgesamt produzierten

Teile dividiert wird.[32]

$$\text{Qualitätsgrad} = \frac{\text{Stueckzahl} - \text{nioStueck}}{\text{Stueckzahl}} \quad (3.4a)$$

In der Literatur können die Bezeichnungen der Werte abweichen, in dieser Arbeit wurde das Berechnungsschema und die Bezeichnungen aus den Unterlagen des Partnerunternehmens verwendet. Eine detailliertere Beschreibung kann unter anderem bei Symestic nachgeschlagen werden.[32] Die Berechnung wird im Data Layer durchgeführt, dessen Umsetzung später in diesem Kapitel erklärt wird.

Übermittelt werden die Daten mithilfe des MQTT-Protokolls. MQTT bedeutet „Message Queuing Telemetry Transport“ und ist ein leichtgewichtiges, offenes Nachrichtenprotokoll, das nach dem Publish and Subscribe Prinzip funktioniert. MQTT wurde speziell für die Kommunikation zwischen Geräten konzipiert, außerdem ist MQTT skalierbar und kann mit mehreren Tausend Datenquellen arbeiten.[33] Ähnlich wie beim Streaming mit Kafka „publishen“ Geräte, Clients genannt, ihre Nachrichten auf ein „Topic“ und senden diese an einen Broker. Der Empfänger Client „subscribed“ auf das Topic und erhält die gesendete Nachricht vom Broker. [34] Es gibt unterschiedliche Provider für die Broker Software. In diesem Anwendungsfall wurde Mosquitto verwendet, eine mögliche Alternative wäre zum Beispiel HiveMQ. Im konkreten Anwendungsfall published die Maschinensteuerung auf das Topic „TEC-002“ und sendet die Nachrichten an einen Broker. Als Broker dient eine IPC die im Werk aufgestellt wurde, mittels eines SAS-Tokens ist der Zugriff darauf eingeschränkt.

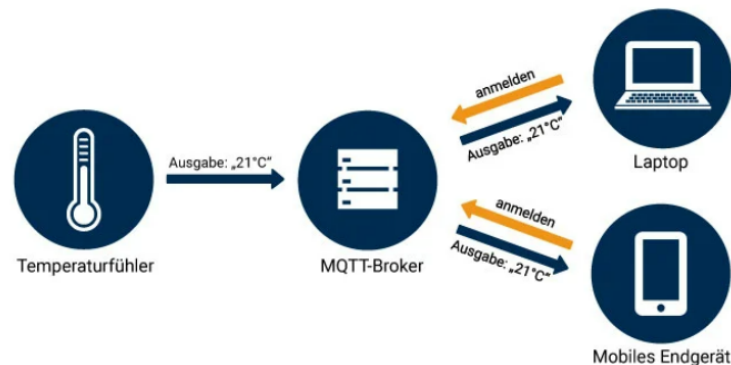


Abbildung 3.12: MQTT Prinzip  
Quelle:OPC Router[34]

Beispiele für alternative Protokolle sind:

- CoAP (Constrained Application Protocol)

- XMPP (Extensible Messaging and Presence Protocol)

## Data Layer

Im folgenden Abschnitt wird die Implementierung des Data Layer des Prototyps erklärt.

### Azure IOT Hub

Der Azure IOT-HUB ist ein Cloud-Service für die Verwaltung von unterschiedlichen IOT-Geräten und kann als Nachrichten-Hub verwendet werden, er ist somit das „Tor in die Cloud“. Der IOT-Hub kann als Broker verwendet werden, hat aber nicht dieselben Eigenschaften wie ein vollwertiger MQTT-Broker, wie Mosquitto oder Hive. Der IOT-Hub steuert den Nachrichtenverkehr innerhalb der Cloud. Durch sogenanntes „Message Routing“ können Nachrichten von IOT-Geräten an Speichercontainer, Datenbanken oder andere Cloud-Dienste wie in diesem Fall den Azure Event-Hub weitergeleitet werden.[35] Um Nachrichten empfangen zu können, muss im IOT-Hub ein neues IOT-Gerät angelegt werden. Nach dem Erstellen werden Verbindungsschlüssel generiert, die benötigt werden, um über das Gerät auf den IOT-Hub zuzugreifen. Der Name des IOT-Gerätes dient als Topic für das Publishen und Subscriben von MQTT-Nachrichten. Mittels der Nachrichtenrouting Funktion werden die Nachrichten an den Azure-Event-Hub weitergeleitet. Zusätzlich bietet der IOT-Hub die Möglichkeit, den Eingang der Nachrichten zu beobachten und zu prüfen. Skalierbarkeit ist hier dadurch gegeben, dass für jedes Gerät, das zukünftig an die Architektur angeschlossen werden soll ein neues IOT-Gerät im IOT-Hub angelegt wird.

### Event-Hub

Der Azure Event-Hub ist ein wichtiger Bestandteil des Data Layers. Der Event Hub bietet die Funktionalitäten einer Streaming-Plattform und agiert in der Azure Cloud als „Event ingestion Service“. [36] So gelangen die Echtzeit-Daten in andere Applikationen, wo sie weiterverarbeitet, transformiert und gespeichert werden können. In der vorliegenden Architektur wird vom API-Layer aus, in einem Databricks Notebook, ein Consumer für die Streaming-Daten aus dem Event Hub angelegt. Für jedes neue Gerät, das angebunden werden soll, wird eine neue Instanz des Event-Hubs erstellt, auf die sich der Consumer, mittels Connectionstring, verbinden kann.

## API-Architektur: Databricks

Für die Implementierung des Stream-Consumers, wird ein Scala-Skript in einem Databricks Notebook verwendet. Databricks ist ein Datenverarbeitungstool, das in der Cloud bereitgestellt wird. Databricks basiert auf Apache Spark und bietet unterschiedliche APIs an, dazu gehören SQL, Python, R und Scala. Das zugrundeliegende Rahmenwerk von Apache Spark ist konzipiert, um große Datenmengen zu verarbeiten. Spark stellt hohe Performance durch Verteilung der Rechenleistung auf mehrere Computer zur Verfügung. Dieser Verbund an Rechnern wird auch Cluster genannt [37]. Spark ist um bis zu 100-mal schneller als die Map-Reduce Workflows von Hadoop und wird für unterschiedliche Einsatzzwecke wie Machine Learning oder, wie in diesem Fall, für ETL-Prozesse verwendet. [38] Die Verbindung zum Event-Hub geschieht mittels eines Connectionstrings. Mithilfe des Connectionstrings wird automatisch auf den richtigen Event-Hub zugegriffen. Die eingehenden Daten werden in einen Dataframe geschrieben, siehe Abbildung 3.13.

	Body ▲
1	{"ts": 1687896199.1795468, "weg": 24, "kraft": 273}
2	{"ts": 1687896199.1795468, "weg": 24, "kraft": 273}
3	{"ts": 1687896206.382604, "weg": 10, "kraft": 289}
4	{"ts": 1687896206.382604, "weg": 10, "kraft": 289}
5	{"ts": 1687897307.8361628, "weg": 19, "kraft": 241}
6	{"ts": 1687897307.8361628, "weg": 19, "kraft": 241}
7	{"ts": 1687897315.6057503, "weg": 38, "kraft": 594}

Abbildung 3.13: Dataframe Stream

Nach Transformation der Nachricht hat der Dataframe die folgende Form:

	Kraft ▲	Timestamp ▲	Weg ▲
1	417	2021-08-25T16:12:07.403+0000	13.530992
2	895	2021-08-25T16:12:07.640+0000	23.218481
3	897	2021-08-25T16:12:07.640+0000	23.253273
4	899	2021-08-25T16:12:07.640+0000	23.261513
5	901	2021-08-25T16:12:07.643+0000	23.290812
6	904	2021-08-25T16:12:07.643+0000	23.3137
7	905	2021-08-25T16:12:07.643+0000	23.332926

Abbildung 3.14: Dataframe Transformiert

Beispiele für mögliche Alternativen zu Databricks/Spark sind:

- Apache Hadoop
- Apache Sqoop
- Presto

### **RDS-Architektur: Delta Tables**

Databricks stellt dem Anwender, mit der Delta Architektur, eine Speicherarchitektur zur Verfügung, welche die Vorteile von Data Lakes und Data Warehouses zu einem „Data Lakehouse“ verknüpft. Die Daten im Data Lakehouse werden in sogenannten Delta Tables gespeichert. Diese Daten liegen im speziellen „Delta“ Dateiformat vor, dessen Performance jene des Apache Spark Standard Formats „Parquet“ um das zehnfache bis hundertfache übertrifft. Weiterhin bieten Delta Tables weitere Funktionen wie die Möglichkeit Abfragen nur über Teile einer Tabelle durchzuführen und die Möglichkeit Batch und Streaming Daten in dieselbe Tabelle zu schreiben. Die wohl wichtigste Funktion des Delta Table ist jedoch die Versionierung. Bei jeder Änderung in einem Delta Table wird eine neue Version desselben angelegt, über einen Zeitstempel sind ältere Versionen jederzeit verfügbar und wiederverwendbar [39]. Weiterhin unterstützen Delta Tables „ACID“ Transformationen. Die Bedeutung von ACID wird hier kurz erläutert [40]:

- Atomicity: Alle Änderungen an einem Table werden auf einmal durchgeführt
- Consistency: Wird der Änderungsprozess nicht ordnungsgemäß durchgeführt, so wird der Ursprungszustand des Tables wiederhergestellt
- Isolation: Transaktionen sind immer nur für den durchführenden User sichtbar
- Durability: Änderungen bleiben nach Abschluss einer Transaktion bestehen und sind nicht rückgängig zu machen. Im Falle eines Fehlers muss eine neue Transaktion durchgeführt werden.

Im betrachteten Anwendungsfall dienen die Delta Tables als Stream-Sink. Das bedeutet, dass eine eingehende Nachricht als Erstes in einen Delta Table geschrieben wird, bevor sie verarbeitet wird. Dies hat den Vorteil, dass alle vergangenen Nachrichten immer verfügbar und nachvollziehbar abgespeichert sind. Für die weitere Transformation der übermittelten Daten wird die jeweilige Zeile aus dem Delta Table in einen Dataframe geladen. Beispiele für mögliche Alternativen zu Delta Tables sind:

- Azure Data Lake
- Amazon Web Service Data Lake
- Apache Kudu

## Data Consumers

In diesem Abschnitt werden die Komponenten des Data Consumer Layer vorgestellt und deren Funktion im POC erklärt.

### PostgreSQL

PostgreSQL ist ein, objektrelationales, open-source Datenbanksystem, das sich eng am SQL-Standard orientiert. Das Datenbanksystem erlaubt komplexe Abfragen, die Definition eigener Datentypen, Funktionen und Operatoren. Wie die meisten Datenbanksysteme arbeitet PostgreSQL nach dem Client-Server Prinzip, bei der ein Server die Verbindungen und die Dateien verwaltet. Der Verbindungsaufbau geschieht mittels einer Software mit User-Interface wie pgAdmin oder über die Kommandozeile mit psql. PostgreSQL bietet einige Erweiterungen und deshalb auch höhere Modularität und Flexibilität [41]. Eine Erweiterung, die für diesen Prototyp genutzt wurde, ist Timescale DB. Die Datenanalyse Anwendung, Databricks, schreibt die Daten aus dem Stream in parallelen Prozessen in drei verschiedene Tabellen in der Datenbank TEC002. Die Rohdaten aus der Prägestation werden zuerst in einen Delta Table und anschließend direkt in die Tabelle „Messwerte“ geschrieben. Die Maschinendaten Zykluszeit, Stillstandszeit, Anzahl gefertigter Artikel und Anzahl produzierter Ausschuss werden in die Tabelle „Maschinendaten“ geschrieben. Der aus den Maschinendaten berechnete OEE-Wert wird in der Tabelle „OEE“ gespeichert. Beispiele für mögliche alternative Datenbanksysteme sind:

- MS SQL
- Maria DB
- Mongo DB

### Timescale DB

Zeitreihendatenbanken sind optimiert für die schnelle, Echtzeit - Speicherung von großen Datensätzen. Durch diese Eigenschaften weisen sie eine besondere Eignung für das Sammeln von Daten aus IOT-Geräten wie Sensoren auf. In Zeitreihendatenbanken werden numerische Werte aus Streaming Daten mit



Zeitstempeln versehen. Diese Datenbanktypen gehören zu den NoSQL-Systemen [42]. Eine Besonderheit von Timescale ist die Verwendung von Hypertables. Hypertables sind PostgreSQL Tabellen die Daten automatisch nach Zeit partitionieren. Dadurch sind die Lese- und Schreibprozesse in Hypertables optimiert, ansonsten lassen sie sich wie normale PostgreSQL Tabellen verwenden.[43] Für die Anwendung des Prototyps wäre die Verwendung von Azure Time Series Insights, in der Cloud, prädestiniert gewesen, allerdings, plant Microsoft diese Ressource im Jahr 2025 auslaufen zu lassen. Deswegen wurde PostgreSQL ausgewählt, da die Datenbank einfach in der Cloud implementiert werden kann und über eine Timescale DB Erweiterung verfügt. Timescale DB unterstützt SQL und die geläufigsten Datentypen für IOT-Geräte. In den meisten Anwendungsfällen sind Float, Integer und String ausreichend. Beispiele für mögliche alternative Datenbanksysteme sind:

- Influx DB
- Open TSDB
- Prometheus

### **Visualisierung: Grafana**

Grafana ist ein Open-Source Visualisierungstool, das Daten aus unterschiedlichen Quellen beziehen und darstellen kann. Grafana ist kompatibel mit unterschiedlichen Datenbanksystemen, wie Prometheus, Influx DB, MS SQL oder PostgreSQL. Für die Visualisierung ist eine Verbindung auf die jeweilige Datenquelle notwendig. Die Software kann lokal oder in der Cloud gehostet werden.[44] In dieser Masterarbeit wurde auf Grafana ein Monitoring Dashboard aufgebaut, das die Sensorwerte aus der Prägestation, die OEE der Anlage und weitere anlagenrelevante Daten anzeigt. Beispiele für mögliche alternative Visualisierungstools sind:

- Influx DB
- Prometheus
- Graphite

### **Prozessklärung zusammengefasst**

Die Maschinensteuerung erhält als Prozessdaten die Messwerte der Sensoren, fügt einen Zeitstempel hinzu und parset diese zu JSON. Als Maschinendaten

werden die Stillstandszeiten, Zykluszeiten, Produzierte Menge und der Ausschuss erhoben, diese Daten werden ebenfalls zu JSON formatiert. Die Messwerte werden dann mittels MQTT-Protokoll an einen Computer gesendet, der im Werk als Broker agiert. Dieser Prozess wurde direkt auf der Steuerung in „strukturierter Sprache“ implementiert. Die Übertragung der MQTT-Nachrichten funktioniert nach dem Publish/Subscribe Prinzip. Der Broker ist dafür zuständig, die Daten an den Data Layer zu übertragen. Für die „ingestion“ der Daten in den Data Layer ist der Azure IOT-Hub zuständig. Innerhalb des Hubs wurde die Anlage als Gerät angelegt, wobei das IOT-Gerät als Topic verstanden werden kann. Nun published die Steuerung die Daten als MQTT verpackt auf das Topic „TEC-002“ und der Broker übernimmt die Verteilung. Als Nächstes gelangen die Daten über den IOT-Hub in den Data Layer und über das Message-Routing in den Event-Hub. Der Stream wird in einen Databricks Cluster gelesen, transformiert und anschließend in einen Delta Table geschrieben. Der Delta Table als Stream-Sink übernimmt die Funktion eines RDS und speichert aktuelle und vergangene Streams ab. Parallel zum Schreibprozess in den Delta Table werden die Messwerte in einer PostgreSQL Datenbank, in der Tabelle „Messwerte“ beziehungsweise „Maschinendaten“ abgespeichert. Ab hier geschieht der Übergang zum „Data Consumers“ Layer. Die PostgreSQL Datenbank wurde ausgewählt, da sie über eine Extension für Timescale DB verfügt, die es ermöglicht Zeitreihendaten effizient zu speichern. Das Stream-Processing wurde mithilfe der Programmiersprache Scala implementiert, Databricks Spark dient als Consumer. Im Gegensatz zu den Prozessdaten werden die Maschinendaten für weitere Berechnungen verwendet. Aus diesen Informationen wird im selben Databricks Notebook die OEE berechnet. Zur Berechnung werden die Daten aus dem RDS in einen neuen Dataframe gelesen. Die Berechnung erfolgt nach den im Abschnitt 3.1 angegebenen Formeln. Die neu berechneten Werte und die Kennzahl OEE bilden einen separaten Dataframe, der in die Tabelle „OEE“ geschrieben wird. Für die Demonstration und Live-Visualisierung der Werte wird Grafana verwendet, das es ermöglicht aus Datenbanken Visualisierungen zu generieren. Hier dienen die einzelnen PostgreSQL-Tabellen: Messwerte, Maschinendaten und OEE als Datenquellen.

## 3.5 Analyse

Nach der Implementierungsphase wurden die Systeme wie unter Abschnitt 3.4.1 definiert, zuerst unabhängig voneinander getestet. Als Erstes wurde die PostgreSQL Datenbank aufgebaut und darin die Tabellen angelegt. Als Nächstes wurde der API-Layer und darin die notwendigen Berechnungen und Transformationen implementiert und getestet. Getestet wurde vorerst mit offline-

batches der realen Maschinendaten. Nach dem verifiziert werden konnte, dass die Analyse funktioniert, wurden die Daten in die Datenbank geschrieben. Diese Daten wurden dann mithilfe einer Grafana Instanz in der Cloud, auf einem Dashboard visuell dargestellt, die Datenbank-Tabellen dienen dabei als Datenquellen. Als Letztes wurde der Streaming-Layer implementiert, was rückblickend den arbeitsintensivsten Prozess der gesamten Entwicklung darstellte. Der Arbeitsaufwand bestand einerseits darin, die Steuerung an den Broker anzubinden, wofür intensive Unterstützung der Werks-IT notwendig war, und andererseits darin, dass der API-Layer von Batch auf Stream-Processing umgestellt werden musste. Erst nachdem die Verbindung der einzelnen Komponenten untereinander funktioniert hat, war es möglich, wie angedacht, den früheren Prototyp mit dem neuen Prototyp zu vergleichen. Abbildung 3.15 zeigt eine Gegenüberstellung der Prototypen auf funktionaler Ebene, anhand der ausgearbeiteten Anforderungen.

Anforderungen	Bestehender Prototyp	Neuer Prototyp
Skalierbarkeit		X
Echtzeitfähigkeit		X
Modularität		X
Verlässlichkeit	X	n/a
Multi-Source	X	X
Visualisierung	X	X

Abbildung 3.15: Vergleich der Prototypen  
Quelle: eigene Darstellung

Kurzum übertrifft die entwickelte Architektur, den alten Prototyp, in allen betrachteten Bereichen. Hervorgehoben werden müssen aber vor allem die drei Eigenschaften: Echtzeitfähigkeit, Skalierbarkeit und Modularität. In der ursprünglichen Anwendung wurden die Informationen parallel in eine lokale und eine serverseitige Datenbank geschrieben. Die Daten wurden aber nur „on demand“ von der lokalen in die Server Datenbank übermittelt. Das heißt, dass teilweise nur einmal pro Woche oder Monat ein Batch-Lauf getriggert wurde. Dabei wurden mehrere 1000 Zeilen übermittelt und in Power BI dargestellt. Durch das Event-getriebene, Stream-Processing werden Zustandsänderungen, Zeile für Zeile übermittelt. Für den API-Layer bedeutet das automatisch eine Verbesserung der Performance, da nicht, wie in der Testphase, ganze Dataframes von bis zu 1000 Zeilen gelesen, transformiert und in die Datenbank geschrieben werden müssen. Auch die Visualisierung in Power BI hat seine Grenzen in Bezug auf Performance. Echtzeitdarstellung ist nur bedingt möglich. In Bezug

auf die Skalierbarkeit der Architektur ist ebenfalls eine Verbesserung feststellbar. Während der Testphase haben sich bereits potenzielle neue Anwendungsfälle innerhalb der Tecton Anlage ergeben, welche hier noch nicht implementiert wurden. Zum Beispiel wurde bereits diskutiert, die Leistungsaufnahme der Servomotoren in der ersten Station „Draht einziehen“ aufzuzeichnen. In diesem Fall wäre es lediglich notwendig im IOT-Hub ein weiteres Gerät anzulegen, die Daten von der Steuerung via MQTT an den Broker auf das Topic des Gerätes zu senden, wobei der Gerätenamen automatisch das Topic ist. In der Cloud übernimmt der IOT-Hub das Routing der Nachrichten an den Event Hub und Databricks subscribed auf das Topic, um die Daten zu laden. In der PostgreSQL Datenbank wäre es notwendig eine neue Tabelle anzulegen, wohin die Daten geschrieben werden und von wo aus Grafana auf die Daten zugreifen kann. Um den Systemansatz aufzugreifen, würde dies eine Erweiterung innerhalb des Systems, der Tecton Anlage, auf der Hardwareseite, bedeuten. Dadurch würde ein weiteres Subsystem, mit dem Beinamen „Station Draht einziehen“ entstehen, welches wiederum über Subsysteme, die Servomotoren verfügt. Da ein neues Gerät möglicherweise Daten von anderer Struktur sendet, ist Modularität ein wichtiges Stichwort. Nach dem Domain-Driven-Design Ansatz muss es möglich sein, die für die Anwendung beste Technologie zur Verfügung zu stellen. Dadurch das Azure unterschiedliche Services zur Verfügung stellt und allein Databricks mehrere Programmier API's unterstützt, ist es möglich unterschiedlicher Module hinzuzufügen, sofern dies notwendig ist. Auch in Bezug auf erweiterte Funktionalität spielt die Modularität eine wichtige Rolle. Das Beispiel der Prägestation wäre prädestiniert für die Verwendung von Machine Learning Algorithmen, um nicht nur das Verhalten zu überprüfen, sondern in weiterer Folge einen eventuellen Ausfall voraussagen zu können.

Da es sich bei der Anwendung um eine Serviceleistung handelt, spielt die Verlässlichkeit eine Rolle. Da die Anwendung aber erst seit Kurzem in Betrieb ist, kann zum Zeitpunkt der Abgabe dieser Arbeit keine genaue Aussage über die Verlässlichkeit getroffen werden.

Eine weitere wichtige Anforderung, die nicht aus der Literaturrecherche hervorging, ist die Zukunftssicherheit der Anwendung. Diese Anforderung entstand aus Diskussionen während der Testphase. Wie in der Einleitung erwähnt, ist die Erhaltung der Wettbewerbsfähigkeit in Hochlohnländern ein Grund für den Vormarsch von Industrie 4.0 und Big-Data Technologien. Da es sehr wahrscheinlich ist, dass deshalb der Einsatz dieser Technologien in Hochlohnländern weiter ansteigt, wird es notwendig sein, immer mehr Maschinen und Anlagen an eine Big-Data-Architektur wie die vorgestellte anschließen zu können. Deshalb be-

kommt der Begriff „Skalierbarkeit“ eine ganz neue Bedeutung. Die skalierbare Datenarchitektur ist damit automatisch nachhaltig und dadurch „zukunftsicher“. Eine wichtige Anforderung sollte somit lauten: „Skalierbarkeit und Zukunftssicherheit“. In Abbildung 3.16 ist grafisch dargestellt, wie die Architektur von außen erweitert werden könnte, wenn mehrere Maschinen und Anlagen angebunden werden, um Zukunftssicherheit zu gewährleisten.

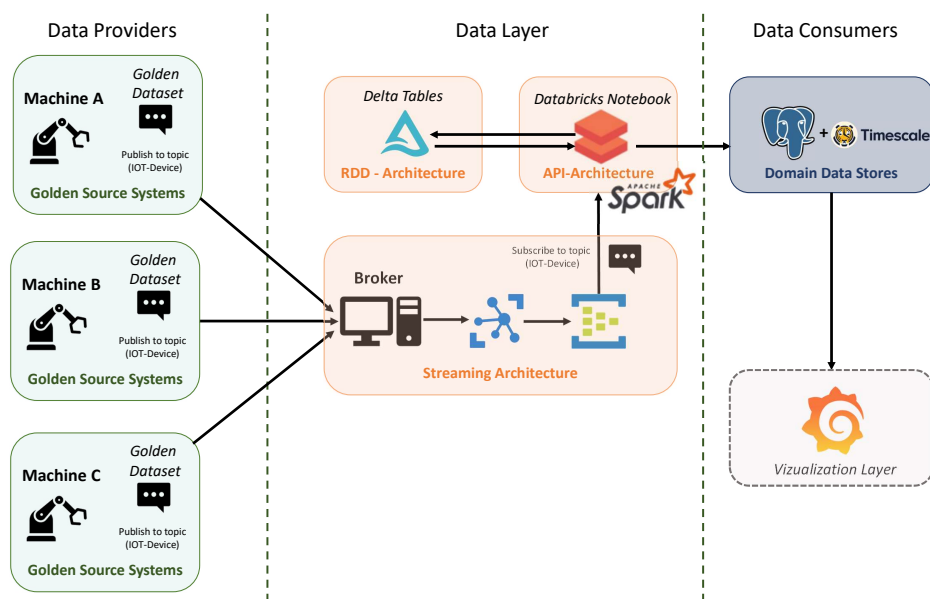


Abbildung 3.16: Erweiterte Architektur  
Quelle: Eigene Darstellung

Einige Screenshots aus der Cloud Anwendung und der Visualisierung in Grafana sind in den Anhängen B und C ersichtlich.

# 4 Abgeleiteter Handlungsleitfaden

Für die Implementierung einer Big-Data-Architektur gibt es vielfältige Möglichkeiten, aber aufgrund der Komplexität des Themas, gibt es kaum standardisierte Vorgehensweisen. Diese These stützt sich auf die Erkenntnis, dass die beste Umsetzung sich immer an den Anforderungen der jeweiligen Domäne orientiert, also Domain-Driven ist. Sehr allgemeine Vorgehensweisen für die Umsetzung von Big-Data-Projekten werden von BITKOM [45] sowie von Meier-Huber und Köhler [46] vorgestellt. Die in dieser Arbeit vorgestellte Vorgehensweise überschneidet sich teilweise mit den Prozessen der vorhin genannten, ist jedoch stark spezifiziert auf das Produktionsumfeld. Angesichts dessen ist der vorgestellte Handlungsleitfaden als Vorschlag, aber nicht als neuer Industriestandard zu verstehen. Der Prozess ist in einem Flussdiagramm am Ende des Kapitels in Abbildung 4.1 dargestellt.

## 1. Ziele und Anwendungsfälle definieren

In der Startphase ist es wichtig, Potenziale für die Anwendung von Big-Data in der Produktion zu finden und zu erkennen. Dies kann zum Beispiel auf der Grundlage einer gegebenen Problemstellung, wie es in dieser Arbeit der Fall war, geschehen. Gibt es keine konkrete Problemstellung, kann dies darauf hinweisen, dass das Bewusstsein über die Potenziale von Big-Data nicht vorhanden ist. In diesem Fall empfiehlt es sich zu analysieren, inwieweit Big-Data-Technologien von fachverwandten Unternehmen genutzt werden. Meier-Huber und Köhler[46] empfehlen in diesem Fall interne Workshops und Umfragen, Recherchen zur Marktsituation, Internetrecherche nach Use-Cases oder Messebesuche. Ein hoher Erfolgsfaktor bietet der Austausch oder das Networking mit anderen Industrieunternehmen, so kann schnell festgestellt werden auf welchem Stand sich das Unternehmen informationstechnisch befindet. Einige Potenziale speziell für produzierende Unternehmen können in den Rechercheergebnissen in Kapitel 2.3.3 nachgeschlagen werden. In weiterer Folge ist es notwendig eine übergeordnete Big-Data-Strategie, aufzusetzen, in der festgehalten wird, wie das Unternehmen in Bezug auf Daten agiert, in der Regel wird diese Strategie von der Führungsebene unter Einbeziehung der Fachex-

perten definiert. Eine durchgängige Strategie hilft auch dabei, Datensilos zu vermeiden. Von hoher Wichtigkeit ist es Ziele zu setzen, die erreichbar und mit den internen Kompetenzen oder mit Unterstützung durch externe Partner umsetzbar sind. Ein Rückschluss dieser Masterarbeit ist die hohe Relevanz der Kenntnisse über die eigenen Prozesse und die Kenntnisse über die unternehmenseigene IT-Infrastruktur, auf Hardware als auch auf Softwareebene. Ist ein möglicher Anwendungsfall gefunden, ist es unbedingt notwendig den Prozess zu analysieren, um diesen im Detail zu verstehen. In den meisten Fällen ist es zu anfangs ausreichend, den Prozess Schritt für Schritt durchzugehen und in einem Flussdiagramm darzustellen. Dabei sollten die Datenquellen (Golden Sources) und Datenkonsumenten sofort sichtbar sein.

## 2. Anforderungen definieren

Schritt Nummer 2 bezieht sich auf die Systemanforderungen. Eine mögliche Einteilung der Anforderungen könnte sein: Anforderungen an Funktion, Anforderungen an Nutzbarkeit, Anforderungen an Hardware und Anforderungen an Software. In diesem Leitfaden wurden die Anforderungen an Hard- und Software als separate Prozessschritte 3 und 4 angesehen. Einige, allgemeingültige Anforderungen, die die produzierende Industrie an Big-Data-Technologien stellt, wurden bereits in Kapitel 2.3.3 dieser Masterarbeit ausgearbeitet. Aus der Definition der Anwendungsfälle und Ziele ergeben sich meistens einige Grundanforderungen an Funktionen, die umgesetzt werden müssen. Daraus abgeleitet lässt sich die Hard- und Software definieren, die diese Anforderungen am besten erfüllt. Für die Definition von Prozess- und Unternehmensspezifischen Anforderungen empfiehlt es sich Workshops mit Fachabteilungen, abzuhalten, da vor allem IT-Security bei der Erhebung und dem Transfer der Daten innerhalb und außerhalb des Unternehmens eine Rolle spielt. IT-Abteilungen können in der Regel auch Support bezüglich benötigter Hard- und Software leisten. Je nachdem wie viele Anforderungen herausgearbeitet werden, kann es ratsam sein diese zu priorisieren.

## 3. Hardwaretauglichkeit

Bei der Analyse der Hardwarelandschaft stellt sich die grundlegende Frage nach der Tauglichkeit der verwendeten Geräte Informationen und Daten zu übertragen. Hier gabelt sich der Prozess zum ersten Mal, und es stellt sich die Frage:

- Ist die notwendige Hardware vorhanden, um Daten zu erheben?

Wenn diese Frage mit Ja beantwortet werden kann, ist der nächste Schritt die Abfrage nach der Softwaretauglichkeit. Auf jeden Fall empfiehlt es

sich, die vorhandene Hardware zu katalogisieren und deren Eigenschaften festzuhalten, falls dies noch nicht geschehen ist. Relevante Eigenschaften für Hardware sind zum Beispiel: Leistung, Kommunikationsschnittstellen (IO-Link, OPC-UA), Formate in denen kommuniziert wird (csv, json, txt), unterstützte Datentypen und Geschwindigkeit. Im Idealfall lassen sich Datenmodelle über die eingesetzte Hardware standardisieren. Wird die Frage mit Nein beantwortet, so ergibt sich daraus eine weitere:

- Welche Hardware ist geeignet, um die Daten zu erheben?

Hier wäre ein Vorschlag über Internetrecherche, Support durch Fachabteilungen (IT, Steuerung und Automatisierungstechnik) oder externe Fachexperten eine Liste mit möglichen Geräten zu erstellen und deren Eigenschaften und Kosten gegenüberzustellen. Speziell im Bereich der Produktion sind solche Hardwarekomponenten: Sensoren, Speicherprogrammierbare Steuerungen oder Mikrocontroller. Die vorhandenen Geräte müssen auch hier nach Anwendungsfall und Eigenschaften katalogisiert werden.

#### 4. Softwaretauglichkeit

Die Tauglichkeitsprüfung der Software folgt dem Prinzip von Schritt 3. Auch hier stellt sich die grundlegende Frage:

- Ist die notwendige Software vorhanden, um Daten aus dem Betrieb zu erheben und zu analysieren?

Wird diese Frage mit Ja beantwortet, kann damit begonnen werden, das Big-Data-Architektur Konzept zu definieren. Vergleiche dazu Schritt 5 und Kapitel 3.3. Die Beantwortung dieser Frage setzt in jedem Fall Kenntnis über die verfügbare IT-Infrastruktur voraus, dies unterstreicht einmal mehr die Wichtigkeit der ganzheitlichen Betrachtung von (digitalen) Prozessen und Tools in einem Unternehmen. Grundlegende, weiterführende Fragen zur Integration der Big-Data-Architektur in die bestehende Infrastruktur könnten sein:

- Ist genügend Kapazität in Speichern vorhanden?
- Ist die Netzwerkstruktur auf große Datenströme ausgelegt?
- Wird bereits eine Cloud verwendet, gibt es schon Cloud basierte Tools, die verwendet werden können?
- Ist Erweiterbarkeit und Skalierbarkeit mit den aktuellen Technologien möglich?

Es wird empfohlen, in jedem Fall eine Landkarte zu erstellen, welche die vorhandenen Technologien zeigt. Ist die notwendige Software nicht vorhanden, stellt sich die Frage:



- Welche Software ist geeignet, für die Umsetzung des Anwendungsfalles?

Diese Frage lässt sich wie die Hardwarefrage unter Punkt 3 durch Internetrecherche, Support durch Fachabteilungen oder externe Fachexperten beantworten. Eine Orientierungshilfe für eine schnell integrierbare Architektur bietet der Prototyp in Kapitel 3.4.2 sowie die vorgestellte Referenzarchitektur von Piethin Strengholt. Wichtige Grundfunktionen der einzelnen Layer sind ebenso in Kapitel 3.3 erklärt.

## 5. **Big Data Konzept erstellen**

Basierend auf den Zielen, Anforderungen, der ausgewählten Hard- und Software kann mithilfe der in Kapitel 3.3 vorgestellten Ideen ein Konzept für die Big-Data-Architektur erstellt werden. Einige allgemeingültige Grundanforderungen an die Big-Data-Architekturen finden sich in den Ergebnissen der qualitativen Inhaltsanalyse in Kapitel 2.3.3. Grundlegende Fragen, die sich bei der Konzeptionierung stellen, könnten sein:

- Muss das System zwingend Echtzeit-fähig sein?
- Batch oder Stream Processing?

Am Anfang ist es sinnvoll eine Visualisierung der Architektur vorzunehmen, dies kann zuerst „High Level“ geschehen, um eine erste Übersicht zu erlangen. Ein weiterer Vorteil der Visualisierung ist, dass sie es ermöglicht, die Grundidee des Konzeptes schnell und anschaulich nach außen zu kommunizieren.

## 6. **Prototyp implementieren**

Sind die Ziele und Anforderungen des Projektes definiert und wurden diese in ein erstes Konzept überführt, kann mit der Implementierung eines Prototyps begonnen werden. Findet die Implementierung im Team statt, sollten regelmäßige Meetings zur Statusabfrage der Arbeitspakete stattfinden. Ebenfalls sollten Stakeholder regelmäßig über den Stand des Projektes informiert werden. Die Verwendung eines Repositories zur Verwaltung und Versionierung der Codes, wird empfohlen. Beispiele für beliebte Dienste sind: GitHub, GitLab oder GitBucket. Nachfolgend wird eine mögliche Vorgehensweise genannt, die sich an der Implementierungsphase des Prototyps dieser Arbeit orientiert. Es muss darauf aufmerksam gemacht werden, dass es sich dabei nicht um eine standardisierte Vorgehensweise handelt. Am Anfang kann die Kommunikationsschnittstelle zur Datenquelle (z. B.: Steuerung) eingestellt und geprüft werden. Am Beispiel dieser Arbeit wurde am Anfang ein IOT-Hub in der Azure Cloud eingerichtet. Anschließend wurde versucht, von der Steuerung Daten via

MQTT-Protokoll über einen Broker an den IOT-Hub zu senden. Dabei steht die Validierung der Verbindung im Vordergrund, diese Vorgehensweise kann entscheidend sein und in der Schlussphase des Projektes Zeit sparen. Denn, was nützen die besten Analyse-Algorithmen, wenn keine Daten empfangen werden können? Nach der Validierung kann die Verbindung aufrechterhalten werden, sofern diese den laufenden Produktionsbetrieb nicht stört. Im nächsten Schritt können die Grundfunktionen wie Transformation, Aggregation, Datenbankabfragen, Lese- und Schreibprozesse sowie die Visualisierung getestet werden. Im Gegensatz zur Startphase sollten die Daten nun unbedingt den Realdaten entsprechen. Die Daten für diese Tests können aus lokalen Batches oder Streams stammen.

## 7. Testen und Evaluieren

Die Implementierungsphase geht beinahe fließend in die Testphase über. Nachdem die Grundfunktionen des Prototyps validiert wurden, kann die Big-Data Architektur an das „Live-System“ angebunden werden. Fehlfunktionen und deren Lösungen werden kontinuierlich dokumentiert. Nach Abschluss der Funktionstests werden die Testergebnisse unter Berücksichtigung der Ziele und Anforderungen in einem Soll-Ist-Vergleich gegenübergestellt, um zu bewerten, ob die Big-Data Architektur den Erwartungen entspricht. Die Testergebnisse sollen im Team geteilt und den Stakeholdern präsentiert werden, die Ergebnisse der Evaluierung bilden die Grundlage für die weitere Vorgehensweise bezüglich Optimierung und Erweiterung der Anwendung.

## 8. Optimieren und Erweitern

In der Optimierungsphase werden die Feedbacks aus den Evaluationsrunden aufgearbeitet. Die einzelnen Punkte werden priorisiert und umgesetzt. Teilweise werden im Projektverlauf oder in der Evaluierungsphase bereits neue Erkenntnisse zur Verbesserung gewonnen und Potenziale für neue Anwendungsfälle entdeckt. Beispiele für Fragen, die sich dabei stellen könnten, sind:

- Wird durch die Anwendung von Big Data Methoden in der Produktion Mehrwert geschaffen?
- Können Prozesse optimiert werden?
- Ist durch die Implementierung eine Steigerung der Effizienz feststellbar?
- Wurde durch die Implementierung mehr Transparenz geschaffen?

Auf welcher Ebene die Erweiterung stattfindet, ist abhängig von der Evaluierung und der übergeordneten Datenstrategie. Am Beispiel des Proto-

typs aus Kapitel 3.4.2 wurde nur eine Station der Tecton Montagelinie betrachtet und angebunden. Im Fall einer angedachten Erweiterung stellt sich die Frage:

- Wie verhält sich das System bei einer Erweiterung?
- Findet die Erweiterung innerhalb oder außerhalb der Systemgrenzen statt?

Die vorgestellte Architektur ermöglicht eine Erweiterung innerhalb und außerhalb der Systemgrenzen. Eine Erweiterung außerhalb der Systemgrenzen wäre zum Beispiel das Anbinden einer weiteren Anlage. Diese Anlage würde Hardwareseitig ein separates System bilden. Softwareseitig müsste die Big-Data Architektur innerhalb der Systemgrenzen um die benötigten IOT-Devices, API's und RDS erweitert werden.

Unter Kollegen und Beteiligten des Implementierungsprojektes wurde der Handlungsleitfaden geteilt und als anwendbar bewertet. Derzeit ist er im „Integrated Management System“ (IMS) der Zumtobel Group, als offizielles und gepflegtes Dokument abgelegt.

# Prozessdarstellung Handlungsleitfaden

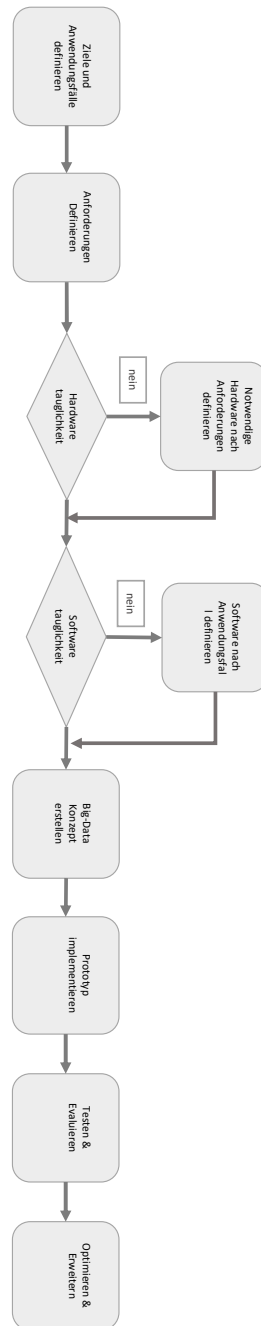


Abbildung 4.1: Handlungsleitfaden  
Quelle: Eigene Darstellung

# 5 Zusammenfassung und Ausblick

In diesem abschließenden Kapitel werden die wichtigsten Methoden und Erkenntnisse zusammengefasst und kritisch betrachtet. Anschließend wird ein Ausblick auf zukünftige Projekte und mögliche interessante Themen für die artverwandte und weiterführende Forschung gegeben.

## 5.1 Zusammenfassung

Die vorliegende Arbeit hat sich mit der Anwendung von Big-Data Architekturen im Bereich der Produktion beschäftigt. Dabei wurden mehrere Themen betrachtet: die Relevanz von Big-Data für die Produktion, die konkrete Umsetzung einer Big-Data Architektur in der Produktion sowie die Organisation von Big-Data Projekten. Dabei wird die Perspektive des Industrial Engineering eingenommen, da dieser Fachbereich oft mit der Umsetzung von Innovations- und Technologieprojekten, im Produktionsumfeld, betraut wird. Die Arbeit wurde in Zusammenarbeit mit der Zumtobel Group durchgeführt. Das Partnerunternehmen verfolgt das Ziel, durch den Einsatz von Datenverarbeitungstechnologien, die bestehende Produktion zu verbessern. Erste Gehversuche mit Datenauswertung in der Produktion hat das Unternehmen an der kritischen Prägestation der Tecton Tragschienenanlage gemacht. Die Umsetzung entspricht allerdings nicht den Erwartungen in Bezug auf Zukunftssicherheit und Erweiterbarkeit. Diese Anlage wurde vom Unternehmen als konkreter Anwendungsfall vordefiniert. Aufbauend auf einem existierenden System soll ein Prototyp für eine Big-Data-Architektur implementiert werden, der in der Produktion möglichst viele Anwendungsfälle abdecken und entsprechend skaliert werden kann. Auf Wunsch des Unternehmens sollte die Datenverarbeitung in der Cloud geschehen. Da das Unternehmen mit der Anwendung von Big-Data in der Produktion zum Startzeitpunkt Neuland betrat, wurde in einem ersten Schritt der theoretische Anwendungsbereich der Arbeit definiert. Dieser liegt im Bereich der Digitalisierung von Produktionsprozessen innerhalb des Fachbereichs Industrie 4.0. In einem ersten Schritt wurden die wichtigsten Technologien und Strategien von Industrie 4.0 sowie der Big-Data Begriff aufgearbeitet. Es wurde

darauf eingegangen, dass Industrie 4.0 Technologien in Verbindung mit Big-Data als Treiber für die Erhaltung der Wettbewerbsfähigkeit von Produktionsunternehmen in Hochlohnländern dienen können. Kurz umrissen bedeutet das aus Sicht der Unternehmen primär die Steigerung der Effizienz, Senkung der Kosten durch Ausfälle oder Qualitätseinbußen und die Prozessoptimierung. Zusammengefasst können diese Ziele durch das Schaffen von Transparenz in der Produktion erreicht werden. Dafür werden Produktionssysteme durch den Einsatz von Sensorik, Mikrocontrollern und Software miteinander verknüpft und kommunikationsfähig gemacht. Diese Geräte werden zu sogenannten „Cyber-Physischen-Produktionssystemen“ und die Fabriken zu „Smart-Factories“. Jene Daten, die die Geräte miteinander austauschen, sollen von Big-Data Architekturen gesammelt, analysiert und visualisiert werden, um eine höhere Transparenz über die Produktionsprozesse zu erlangen. Um der Ausarbeitung des Prototyps einen gewissen Grad an Allgemeingültigkeit zu verleihen, wurden in einer Literaturrecherche die Anforderungen, Herausforderungen, relevanten Daten und Potenziale, welche die Industrie in Bezug auf Big-Data sieht, aufgearbeitet. Als Quellen wurden Fachzeitschriften, Webseiten und facheinschlägige Literatur verwendet, die sich sowohl mit Big-Data als auch mit Industrie 4.0 und deren Verknüpfungen beschäftigt. Die Rechercheergebnisse wurden mithilfe der Methode der „qualitativen Inhaltsanalyse“ kategorisiert. Die Ergebnisse wurden nicht nur aus technischer Sicht, sondern auch aus organisatorischer Sicht betrachtet. Diese Ergebnisse dienten als Basis für die Implementierung einer prototypischen Big-Data Architektur, einerseits und für den später erstellten Handlungsleitfaden, zur Umsetzung von Big-Data Projekten, andererseits. In Kapitel 3.4 wird das Konzept einer prototypischen Big-Data Architektur vorgestellt, die als Proof-Of-Concept dient. Die Vorgehensweise entsprach dabei der Methode „Design Science for Information System Research“[31] bei der ein entwickeltes Artefakt im Zentrum der Forschung steht. Die angewandte Vorgehensweise entspricht dem Modell für „zielorientierte“ Forschung. Vor der Umsetzung wurde der bereits bestehende Prototyp analysiert und auf dessen Schwachstellen hingewiesen. Danach wurde ein Konzept erstellt, das die zuvor ausgearbeiteten Anforderungen erfüllt. Als Basis für das Konzept diente die Referenzarchitektur, die von Piethein Strengtholt in „Data Management At Scale“[28] vorgeschlagen wird. Dieses drei-Layer-Prinzip wurde in Abschnitt 3.3 erläutert. Anschließend wurde ein Konzept nach diesen Ideen aufgesetzt und die verwendeten Technologien für die einzelnen Komponenten und Implementierungsmöglichkeiten definiert. Die Anforderungen an die umgesetzte Architektur stammen aus der Vorgabe, des Unternehmens eine Cloud-basierte Lösung zu entwickeln und aus den Ergebnissen der Literaturrecherche. Nach der Konzeptionierung wurde mit der konkreten Implementierung begonnen. Da

die Zumtobel Group bereits über eine Azure-Subscription verfügt, wurde die Azure Cloud und die darin zur Verfügung gestellten Ressourcen für die Umsetzung verwendet. Für die Bewertung der Architektur wurde diese auf funktionaler Ebene mit dem bereits existierenden Vorgänger-Prototypen verglichen. Aus den während der Umsetzung gemachten Erfahrungen und den aus der Recherche gewonnenen Informationen, wurde abschließend ein Handlungsleitfaden für die Umsetzung von Big-Data-Projekten erstellt. Dieser Handlungsleitfaden umfasst acht Phasen und hat keinen Anspruch auf Allgemeingültigkeit. Ziel war es, eine technologische und organisatorische Orientierungshilfe für die zukünftige Umsetzung von Big-Data Projekten zu bieten.

## 5.2 Fazit

Für das Fazit der Masterarbeit ist es sinnvoll, noch einmal einen Blick auf die am Anfang, definierten Ziele zu werfen:

- Ausarbeiten von Anforderungen an Big-Data Architekturen in der Produktion
- Entwicklung eines Prototyps, der die Anforderungen erfüllt
- Entwicklung eines Handlungsleitfadens für die Umsetzung zukünftiger Big-Data-Projekte

### **Ausarbeiten von Anforderungen an Big-Data Architekturen in der Produktion**

Die Ausarbeitung der Anforderungen, welche die produzierende Industrie an Big-Data Architekturen stellt, wurde in Abschnitt 2.3 im Zuge einer Literaturrecherche mit anschließender Kategorisierung durch die qualitative Inhaltsanalyse durchgeführt. Im Zuge dieser Recherche wurden aber nicht nur Anforderungen, sondern auch Herausforderungen definiert. Ein wichtiger Bestandteil der Recherche ist auch die Suche nach den Daten, deren Auswertung in Bezug auf Mehrwertgenerierung relevant sind. Die wichtigsten Anforderungen sind jedoch: Die Fähigkeit in Echtzeit Informationen abrufen zu können und die Möglichkeit, die Anwendung zu skalieren, sprich auf Veränderung der anfallenden Datenmengen reagieren zu können. Aus Hardwaresicht bedeutet Skalierbarkeit das Hinzufügen neuer Datenquellen, da sich dadurch der Datenstrom erhöht. Die Verlässlichkeit des Systems ist unabdingbar, die Anwendung muss die Funktionen, die ihr von der Anwenderseite gestellt werden, erfüllen und stabil betrieben werden können. Modularität ist eine Eigenschaft der Architektur, die vor allem für die Analyse und Auswertung eine Rolle spielt, es muss

möglich sein unterschiedliche Technologien innerhalb einer Architektur verwenden zu können. Damit sind unterschiedliche Programmiersprachen (APIs), aber auch unterschiedliche Speichertechnologien (Blob Storage, Data Lakes) und Visualisierungstools gemeint. Modularität und Skalierbarkeit weisen eine gewisse Schnittmenge mit der Multi-Source Fähigkeit auf. Es muss grundlegend gewährleistet sein, dass neue Datenquellen hinzugefügt werden können. Neue Quellen haben eventuell abweichende Eigenschaften und Datenmodelle, worauf die Anwendung reagieren können muss. Wo möglich sollten die Datenmodelle, auf denen die Kommunikation der CPS (wie Sensoren) aufbaut, vereinheitlicht werden. In der Praxis ist dies gegeben durch unterschiedliche Hersteller nicht immer machbar. Visualisierung spielt eine Rolle, um Informationen und komplexe Zusammenhänge „fassbar“ zu machen und anschaulich darzustellen. Visualisierung spielt vor allem in der Kommunikation nach außen eine große Rolle. Durch die Literaturrecherche wird deutlich, dass verschiedene Quellen oft dieselben Anforderungen aufzeigen, die oft sehr allgemeinen Charakters sind. Demzufolge sind die ausgearbeiteten Anforderungen auch kritisch zu betrachten. Viele Allgemeine Anforderungen sind vor allem beim Start eines Projektes hilfreich, um eine theoretische Basis zu schaffen. Diese Anforderungen sind deshalb als Orientierungshilfe zu sehen. Von Strengolt[28] wird an mehreren Stellen auf die Wichtigkeit des „Domain-Driven-Design“ Prinzips hingewiesen, dies bezieht sich meistens auf die Auswahl der richtigen Technologie für eine spezielle Aufgabe. Der „Domain-Driven-Design“ Ansatz ist aber auch in der Definition von Funktionen und Anforderungen zu beachten, da diese je nach Domäne abweichen können. Je tiefer man in eine Anwendung eintaucht und, umso länger diese in Betrieb ist, umso mehr spezifische Anforderungen werden sich ergeben. Im Fall des vorgestellten Prototyps bedarf es einer längeren Laufzeit, um Anpassungen feststellen zu können die über die geforderten Grundfunktionen hinausgehen.

### **Entwicklung eines Prototyps, der die Anforderungen erfüllt**

Der entwickelte Prototyp sowie die verwendeten Technologien orientieren sich stark an den Konzepten von Piethin Strengolt[28]. Das in Abschnitt 3.4 vorgestellte Konzept baut auf dessen Referenzarchitektur auf und erfüllt sowohl die recherchierten Anforderungen als auch die Forderung der Zumtobel Group nach einer cloudbasierten Lösung. Die wichtigsten Eigenschaften wie Echtzeitfähigkeit, Modularität und Skalierbarkeit sind in jedem Fall gegeben. Eine neue Station der Tecton Tragschienenanlage kann relativ einfach eingebunden werden. Dafür ist es lediglich notwendig im IOT-Hub ein neues Gerät (Topic) anzulegen und die Daten aus der Anlagensteuerung, über den Broker, an das Topic zu publizieren. Der IOT-Hub routet die Nachrichten automatisch an den Event Hub



weiter, somit ist auch Echtzeitfähigkeit gegeben. Innerhalb der Cloud muss auf der PostgreSQL Datenbank eine Tabelle für das Gerät angelegt werden und auf Databricks ein neuer Delta Table. Durch die geleistete Vorarbeit gibt es dafür schon Vorlagen, die die Erweiterung erleichtern. Auch außerhalb der Systemgrenzen der Tecton Anlage, könnte durch den Anschluss an den Broker, eine Erweiterung erfolgen. Hier gilt es, die Datenmodelle der Sender zu prüfen, eventuell ist eine andere Art der Transformation und Aggregation notwendig. Bezüglich der eingesetzten Technologien gibt es eine hohe Anzahl an unterschiedlichen Möglichkeiten. So liefert etwa eine Online-Suche mit dem Thema: „Big-Data Architektur für Echtzeit Datenstreaming“ eine sehr hohe Anzahl an unterschiedlichen Implementierungsvarianten. Das soll heißen, dass die Umsetzungsmöglichkeiten einer Big Data Architektur genauso vielfältig wie deren Anforderungen sind, was auch der hohen Anzahl an unterschiedlichen Technologien und Anbietern geschuldet ist. Hier gilt die wieder das „Domain-Driven-Design“ Prinzip: Die beste Lösung für den gegebenen Anwendungsfall auswählen! Hilfreiche Quellen für die Auswahl der Technologien und Beispielprojekte sind unter anderem die Onlinedokumentationen von MS-Azure, Github, Towards Data Science oder Medium. Die Technologieauswahl für den POC in dieser Masterarbeit basiert auf den recherchierten Anforderungen und dem Wunsch des Unternehmens eine Cloud Lösung, in Azure bereitzustellen. Außerdem wurden Technologien ausgewählt, die schnell zu implementieren sind und mithilfe derer sich schnell Ergebnisse erzielen lassen.

### **Entwicklung eines Handlungsleitfadens für die Umsetzung zukünftiger Big-Data-Projekte**

Der Handlungsleitfaden, vorgestellt in Kapitel 4, baut auf den Erfahrungen, die während der Umsetzungsphase dieser Arbeit gemacht wurden auf. Für den Handlungsleitfaden gilt nicht der Anspruch nach Allgemeingültigkeit, er stellt keinen neuen Industriestandard dar, sondern bietet lediglich eine Orientierungshilfe. Ein Anspruch, den der Handlungsleitfaden sehr wohl stellt, ist, dass dieser ohne wissenschaftlichen Hintergrund verwendet werden kann. So sollen zum Beispiel die Ziele und Anforderungen definiert werden können, ohne dass wissenschaftliche Methoden wie eine qualitative Inhaltsanalyse eingesetzt werden müssen. Der Handlungsleitfaden soll Unternehmen und deren Mitarbeitern ein Phasenmodell zur Orientierung in der Umsetzung eines Big-Data Projektes bieten. Gemäß dem Untertitel dieser Arbeit, „ein Ansatz aus dem Industrial Engineering“ ist der Handlungsleitfaden vorwiegend an diese Zielgruppe gerichtet. Da dieser Fachbereich in vielen Unternehmen, wie der Zumtobel Group, Innovationsprojekte im Produktionsumfeld leitet. Aufgrund der Zunahme der Nutzung an IT und Big-Data-Technologien (Kapitel 2.2Abbildung 2.3), ist es

wahrscheinlich, dass das Industrial Engineering zukünftig stärker mit diesen Themen konfrontiert sein wird. Aus diesem Grund wurde versucht, den Handlungsleitfaden so zu verfassen, dass dieser für Nicht-Informatiker anwendbar und verständlich ist. Zum Stand Juli 2023 ist es noch nicht möglich festzustellen, wie gut sich der Handlungsleitfaden im Praxiseinsatz bewährt, eine ausführliche Bewertung wird nach Abschluss von zukünftigen Big-Data-Projekten im Werk erfolgen.

## 5.3 Ausblick

Bei der Ausarbeitung des Prototyps als Proof-Of-Concept wurden die Ziele und Vorgaben der Zumtobel Group in den Vordergrund gestellt. Zukünftig plant das Unternehmen, Big-Data nicht nur in der Produktion, sondern unternehmensweit zu nutzen. Mit diesem Projekt wurde der Grundstein dafür gelegt, dem Thema gegenüber mehr Aufmerksamkeit zu erregen und in weiterer Folge auch Akzeptanz zu schaffen. Unter dem Namen „Digital Factory“ werden unterschiedliche Digitalisierungs-Projekte innerhalb der Gruppe, vor allem am Standort Dornbirn, verfolgt. Zukünftig soll die Architektur gemäß ihren Eigenschaften auf mehrere Maschinen und Stationen hoch-skaliert werden. So könnte zukünftig eine IOT-Plattform entstehen, die mehrere Zwecke erfüllt. Führt man diesen Gedanken weiter, wird sich die Frage stellen, welche Potenziale Big-Data abseits der Produktion bietet? Eine Thematik für zukünftige Forschungsfelder wäre die Verknüpfung von „Knowledge-Based-Engineering“ mit Big-Data. Eventuell ist es zukünftig möglich aus den Maschinen, Prozess und Produktdaten Konstruktionsregeln für Maschinen und Anlagen abzuleiten. Dadurch würden auch weitere Unternehmensbereiche, wie die Forschung, Entwicklung, Anlagenbau, Projektierung oder Einkauf profitieren. Aus betriebswirtschaftlicher Sicht ergäben sich dadurch weitere Potenziale zur Kostensenkung und Optimierung. Die Potenziale von Big-Data sind also genauso groß und komplex wie die anfallenden Datenmengen, wenn nicht sogar größer.

# Literatur

- [1] Piethein Strengtholt. „ABN AMRO’s Data Integration Architecture,“ Medium. (2021), Adresse: <https://piethein.medium.com/abn-amros-data-integration-architecture-3d266a59fbdd>.
- [2] C. Brecher, Hrsg., *Integrative Produktionstechnik für Hochlohnländer*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2011.
- [3] H. Kagermann, W. Wahlster und J. Helbig, „Deutschlands Zukunft als Produktionsstandort sichern,“ Berlin, Abschlussbericht des Arbeitskreises Industrie 4.0, 2013.
- [4] B. Vogel-Heuser, T. Bauernhansl und M. ten Hompel, *Handbuch Industrie 4.0 Band 1*, 2. Aufl., 4 Bde. Springer, 2017, Bd. 1, ISBN: 978-3-662-45278-3.
- [5] BITKOM, „Industrie 4.0 - Volkswirtschaftliches Potenzial für Deutschland,“ *Controlling*, Jg. 27, Nr. 8, S. 515–517, 2015, ISSN: 09350381. DOI: 10.15358/0935-0381-2015-8-9-515. (besucht am 18.05.2023).
- [6] V. Emmrich und M. Döbele, *Geschäftsmodell-Innovation durch Industrie 4.0*. Fraunhofer IPA, 2015.
- [7] P. Gölzer, „Big Data in Industrie 4.0 - Eine strukturierte Aufarbeitung von Anforderungen, Anwendungsfällen und deren Umsetzung,“ Diss., Friedrich-Alexander-Universität Erlangen-Nürnberg, Nürnberg, 2016.
- [8] Hottinger, Brüel und Kjaer. „Mittler zwischen realer Welt und Cyberspace: Cyber-Physische Systeme, der Kit der Industrie 4.0.“ (), Adresse: <https://www.hbm.com/de/6267/mittler-zwischen-realer-welt-und-cyberspace-cyber-physische-systeme-der-kit-der-industrie-40/>.
- [9] Hottinger, Brüel und Kjaer. „Alles kommuniziert: Das Internet der Dinge, Fundament der Industrie 4.0.“ abgerufen am: 28.05.2023. (), Adresse: <https://www.hbm.com/de/6264/das-internet-der-dinge-fundament-der-industrie-40/>.

- [10] Alexander Thamm. „Smart Factory – wie die richtige Datenstrategie die vernetzte Produktionsumgebung zum Erfolg führt.“ abgerufen am: 29.05.2023. (18. Aug. 2021), Adresse: <https://www.alexanderthamm.com/de/blog/smart-factory-in-der-industrie-4-0/>.
- [11] Christoph Plass, *Industrie 4.0 als Chance begreifen* (Digitalisierung). Unity, 2015, ISBN: 978-3-946184-03-4. Adresse: [https://www.unity.de/fileadmin/Insights/OPPORTUNITY/OPPORTUNITY\\_Industrie\\_4.0\\_als\\_Chance\\_begreifen.pdf](https://www.unity.de/fileadmin/Insights/OPPORTUNITY/OPPORTUNITY_Industrie_4.0_als_Chance_begreifen.pdf).
- [12] Eberhard Heins. „Industrie 4.0 erfordert intelligente Vernetzung,“ IT-Matchmaker. (2017), Adresse: <https://news.it-matchmaker.com/industrie-4-0-erfordert-intelligente-vernetzung/>.
- [13] Steffen Wulf. „Horizontale Integration – das ganze Unternehmen profitiert von der Smart Factory,“ Industry-Of-Things. abgerufen am: 03.06.2023. (2022), Adresse: <https://www.industry-of-things.de/horizontale-integration-das-ganze-unternehmen-profitiert-von-der-smart-factory-a-acfdebbc8f312893765d6292396af5c3/>.
- [14] COPADATA. „Horizontale und vertikale Integration in einer Smart Factory,“ COPADATA. abgerufen am: 03.06.2023. (), Adresse: <https://www.copadata.com/de/branchen/horizontale-vertikale-integration/#:~:text=Mit%20horizontaler%20Integration%20ist%20die,bis%20ins%20Management%20des%20Unternehmens..>
- [15] S. Hoppe, „Standardisierte horizontale und vertikale Kommunikation,“ in *Handbuch Industrie 4.0*, B. Vogel-Heuser, T. Bauernhansl und M. Ten Hompel, Hrsg., Berlin, Heidelberg: Springer Berlin Heidelberg, 2015, S. 1–20, ISBN: 978-3-662-45537-1. DOI: 10.1007/978-3-662-45537-1\_66-1. Adresse: [http://link.springer.com/10.1007/978-3-662-45537-1\\_66-1](http://link.springer.com/10.1007/978-3-662-45537-1_66-1) (besucht am 08.07.2023).
- [16] Laurenz Wuttke. „Was ist Big Data? Definition, 4V’s und Technologie,“ Datasolut. abgerufen am: 06.06.2023. (), Adresse: <https://datasolut.com/was-ist-big-data/>.
- [17] Oracle. „Was versteht man unter Big Data?“ Was versteht man unter Big Data? abgerufen am: 01.06.2023. (), Adresse: <https://www.oracle.com/de/big-data/what-is-big-data/>.
- [18] Doug Laney, *3D Data Management: Controlling Data Volume, Velocity, and Variety*. Stamford: META Group Inc, 2001, Bd. Application Delivery Strategies.

- [19] Udo Kuckartz, *Qualitative Inhaltsanalyse. Methoden, Praxis, Computerunterstützung*. Weinheim: Beltz Verlagsgruppe, 2016, ISBN: 978-3-7799-4386-0.
- [20] P. Mayring, *Qualitative Inhaltsanalyse - Grundlagen und Techniken*, 12. Aufl. Weinheim: Beltz Verlagsgruppe, 2015, ISBN: 978-3-407-29393-0.
- [21] Smartblick. „Maschinendatenerfassung: Das A und O für höhere Effizienz,“ Smartblick. abgerufen am: 06.06.2023. (), Adresse: <https://smartblick.de/magazin/maschinendatenerfassung-wiki/>.
- [22] Olga Schmid. „Maschinendatenerfassung und Betriebsdatenerfassung im Vergleich,“ FORCAM. abgerufen am: 30.05.2023. (2021), Adresse: <https://forcam.com/maschinendatenerfassung-und-betriebsdatenerfassung-im-vergleich/>.
- [23] OPC-Router. „Maschinendatenerfassung in Fertigung und Produktion,“ OPC-Router. abgerufen am: 30.05.2023. (), Adresse: <https://www.opc-router.de/maschinendatenerfassung/>.
- [24] L. Wang und C. A. Alexander, „Big data in design and manufacturing engineering,“ *American Journal of Engineering and Applied Sciences*, Jg. 8, Nr. 2, S. 223–232, 1. Feb. 2015, ISSN: 1941-7020. DOI: 10.3844/ajeassp.2015.223.232. Adresse: <http://thescipub.com/abstract/10.3844/ajeassp.2015.223.232> (besucht am 11.06.2023).
- [25] Vorne. „Overall Equipment Effectiveness.“ abgerufen am: 30.05.2023. (), Adresse: <https://www.oee.com/>.
- [26] M. Khan, X. Wu, X. Xu und W. Dou, „Big data challenges and opportunities in the hype of industry 4.0,“ in *2017 IEEE International Conference on Communications (ICC)*, Paris, France: IEEE, Mai 2017, S. 1–6, ISBN: 978-1-4673-8999-0. DOI: 10.1109/ICC.2017.7996801. Adresse: <http://ieeexplore.ieee.org/document/7996801/> (besucht am 21.05.2023).
- [27] Martin Kleppmann, *Designing Data-Intensive Applications*. Sebastopol: O’Reilly, 2017, ISBN: 978-1-4493-7332-0.
- [28] P. Strengholt, *Data Management at Scale*, 1. Aufl. O’Reilly, 2020, ISBN: 978-1-4920-5478-8.
- [29] Alex Bekker. „Die „beängstigende“ Sieben: Herausforderungen von Big Data und deren Lösungswege,“ Science Soft. abgerufen am: 07.06.2023. (2018), Adresse: <https://www.scnsoft.de/blog/sieben-herausforderungen-von-big-data>.



- [40] IBM. „ACID-Eigenschaften für Transaktionen.“ abgerufen am: 21.06.2023. (19. Juni 2023), Adresse: <https://www.ibm.com/docs/de/cics-ts/5.4?topic=processing-acid-properties-transactions>.
- [41] „PostgreSQL: Das objektrelationale Datenbank-Managementsystem unter der Lupe,“ Digital Guide IONOS. abgerufen am: 25.06.2023. (14. Sep. 2022), Adresse: <https://www.ionos.at/digitalguide/server/knowhow/postgresql/>.
- [42] Thomas Joos. „Zeitreihendatenbanken für das Speichern von Messdaten,“ Sensoren- und Internet-of-Things-Daten zuverlässig speichern. abgerufen am: 04.06.2023. (15. Jan. 2021), Adresse: <https://www.storage-insider.de/zeitreihendatenbanken-fuer-das-speichern-von-messdaten-a-1034371/>.
- [43] „About Hypertables,“ Timescale Documentation. abgerufen am: 11.06.2023. (), Adresse: <https://docs.timescale.com/use-timescale/latest/hypertables/about-hypertables/>.
- [44] „Open-Source-Tool Grafana - Eine Einführung.“ abgerufen am: 04.06.2023. (27. Mai 2021), Adresse: <https://community.mz-connect.com/blog/open-source-tool-grafana-eine-einfuehrung/>.
- [45] BITKOM, *Management von Big-Data-Projekten*. Berlin: BITKOM, 2013. Adresse: <https://www.bitkom.org/sites/default/files/file/import/130618-Management-von-Big-Data-Projekten.pdf>.
- [46] Mario Meir-Huber und Martin Köhler, „Best Practice für Big Data Projekte - Leitfaden aus Big Data in Austria,“ *#Big Data in #Austria*, 2014. DOI: 10.13140/2.1.3500.8001. Adresse: [https://www.researchgate.net/publication/266819422\\_Best\\_Practice\\_fur\\_Big\\_Data\\_Projekte\\_-\\_Leitfaden\\_aus\\_Big\\_Data\\_in\\_Austria](https://www.researchgate.net/publication/266819422_Best_Practice_fur_Big_Data_Projekte_-_Leitfaden_aus_Big_Data_in_Austria).

# Anhang

## A Qualitative Inhaltsanalyse Strukturblatt

Quelle	Autor	Jahr	Verlag/Herausgeber	ISBN/URL	Ort	Band	Auflage

Selektionskriterien	A	B	C	D
Zutreffend?				

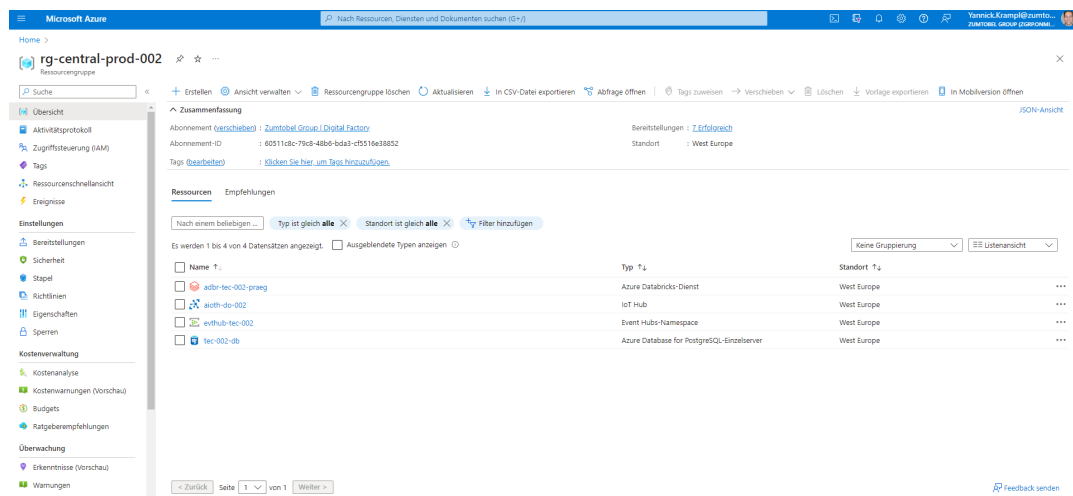
Forschungsfragen	
Was bedeutet Big Data in der Produktion/Industrie?	
Welche Daten sind relevant?	
Welche Eigenschaften haben diese Daten?	
Welche Anforderungen an Big Data Architekturen werden genannt?	
Herausforderung in der Implementierung?	
Potentiale?	



# B Azure Cloud Ansicht der Ressourcengruppe

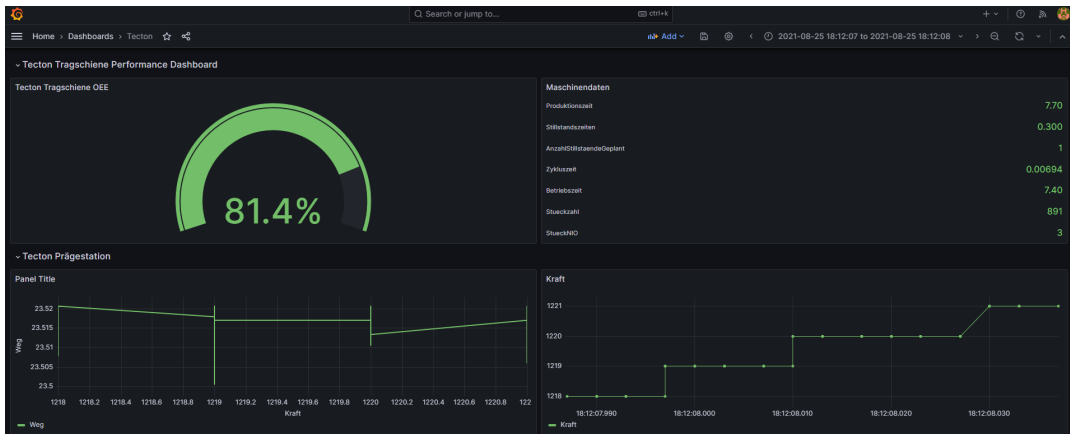
Übersicht der Ressourcengruppe auf Azure, ersichtlich sind die verwendeten Ressourcen:

- Azure IOT Hub
- Azure Event Hub
- Azure Databricks
- PostgreSQL



# C Tecton Dashboard - Grafana

Das Tecton Dashboard besteht aus zwei Reihen, die aufgeklappt werden können. Oben befindet sich die Reihe mit Maschinendaten, hier werden Daten wie gefertigte Stück und OEE angezeigt.



In der unteren Reihe werden die Sensordaten, als Kraft und Weg Diagramme angezeigt.



# Eidesstattliche Erklärung

Ich erkläre hiermit an Eides statt, dass ich die vorliegende Masterarbeit selbstständig und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Die aus fremden Quellen direkt oder indirekt übernommenen Stellen sind als solche kenntlich gemacht. Die Arbeit wurde bisher weder in gleicher noch in ähnlicher Form einer anderen Prüfungsbehörde vorgelegt und auch noch nicht veröffentlicht.

Dornbirn, am Samstag, 29.07.2023

Yannick Peter Karl Krampfl