

**FH Aachen**

**Faculty**

**Electrical engineering and information technology**

**Bachelor Thesis**

**Design and Implementation of a Performance Measurement System  
for an Industrial Sewing Machine**

**Nicolas Harrje**

**Matr.-Nr.: 3518047**

**Referent: Prof. Dr-Ing. ...**

**Korreferent: Prof. Dr-Ing. ...**

**July 3, 2025**

# Erklärung

Ich versichere hiermit, dass ich die vorliegende Arbeit selbstständig verfasst und keine anderen als die im Literaturverzeichnis angegebenen Quellen benutzt habe.

Stellen, die wörtlich oder sinngemäß aus veröffentlichten oder noch nicht veröffentlichten Quellen entnommen sind, sind als solche kenntlich gemacht.

Die Zeichnungen oder Abbildungen in dieser Arbeit sind von mir selbst erstellt worden oder mit einem entsprechenden Quellennachweis versehen.

Diese Arbeit ist in gleicher oder ähnlicher Form noch bei keiner anderen Prüfungsbehörde eingereicht worden.

Aachen, July 3, 2025

## Geheimhaltung - Sperrvermerk

Die vorliegende Arbeit unterliegt bis [Datum] der Geheimhaltung. Sie darf vorher weder vollständig noch auszugsweise ohne schriftliche Zustimmung des Autors, des betreuenden Referenten bzw. der Firma [Firmenname und -sitz] vervielfältigt, veröffentlicht oder Dritten zugänglich gemacht werden.

# Contents

<b>1. Introduction</b>	<b>4</b>
1.1. Motivation und Aufgabenstellung.....	4
1.2. Vorgehensweise .....	5
<b>2. Foundations</b>	<b>6</b>
2.1. Definitions.....	6
2.1.1. IoT and IIoT .....	6
2.2. State of the Art .....	6
2.2.1. Industrial IoT Architectures and Patterns .....	6
2.2.2. Performance measurement in production environments .....	8
2.2.3. IoT-Plattformen .....	8
2.2.4. Differences between Relational and Timeseries Databases .....	9
2.2.5. Dashboarding.....	9
<b>3. Requirements Analysis</b>	<b>10</b>
3.0.1. Functional Requirements .....	10
3.0.2. Non-Functional Requirements.....	10
3.0.3. Constraints .....	10
3.0.4. KPI Selection and Justification .....	10
<b>4. Zusammenfassung und Ausblick</b>	<b>11</b>
<b>Quellenverzeichnis</b>	<b>12</b>
<b>Abkürzungsverzeichnis</b>	<b>13</b>
<b>Abbildungsverzeichnis</b>	<b>14</b>
<b>Tabellenverzeichnis</b>	<b>15</b>
<b>Anhang</b>	<b>15</b>
<b>A. Quellcode</b>	<b>16</b>
<b>B. Rohdatenvisualisierungen</b>	<b>17</b>

# **1. Introduction**

## **1.1. Motivation und Aufgabenstellung**

## **1.2. Vorgehensweise**

## 2. Foundations

### 2.1. Definitions

#### 2.1.1. IoT and IIoT

The term Internet of Things was first coined by [?] when explaining the idea of combining RFID with the internet in an executive meeting. He explains that on the "normal" internet, most of the content is created by human beings. In contrast to this in the Internet of Things the data is generated by things and often describes things. But he emphasizes lays more on the description of things. For example to track and count them. The information to do so would come from sensors and RFID, he says. Of course in these days more of the information on the internet is generated by bots and AI. But other than that the distinction still holds true.

The Internet Society [Rose et al., ] further explains that in the Internet of Things, machines are communicating with each other and are addressable via an own IP address. This standardizes the way in which devices communicate. They also mention that "Today, the Internet of Things has become a popular term for describing scenarios in which Internet connectivity and computing capability extend to a variety of objects, devices, sensors, and everyday items."

The Industrial Internet of Things is just the description of a domain where the IoT is used. In this case in manufacturing. [Wha, ]

### 2.2. State of the Art

#### 2.2.1. Industrial IoT Architectures and Patterns

Due to the requirement that the solution be developed utilizing IoT technologies and is set within a production context, a review of Industrial IoT (IIoT) architectures and patterns was conducted. The Industrial Internet Reference Architecture (IIRA) [Young, 2022] serves as a comprehensive framework, offering valuable insights into various architectural models and design patterns relevant to this domain. This reference architecture describes the following patterns: IoT Component Capability Pattern, Three-Tier Architecture Pattern, Gateway-Mediated Edge Connectivity and Management architecture pattern, Digital Twin Core as a Middleware Architecture Pattern, Layered Databus Architecture Pattern, System-of-Systems Orchestrator Architecture Pattern. Of these patterns only the first two are applicable within the scope of this work. Therefore the other ones will only be described on the surface.

**Architecture Patterns** IoT architecture patterns define the structure and operation of various IoT systems, detailing their implementation and highlighting their unique characteristics.

**IoT Component Capability Model Pattern** A single component and its associated capabilities are described, with the possibility that a component may comprise multiple sub-components. Consequently, the entire system can also be regarded as a component. The specific meanings of the capabilities are illustrated in the accompanying figure.

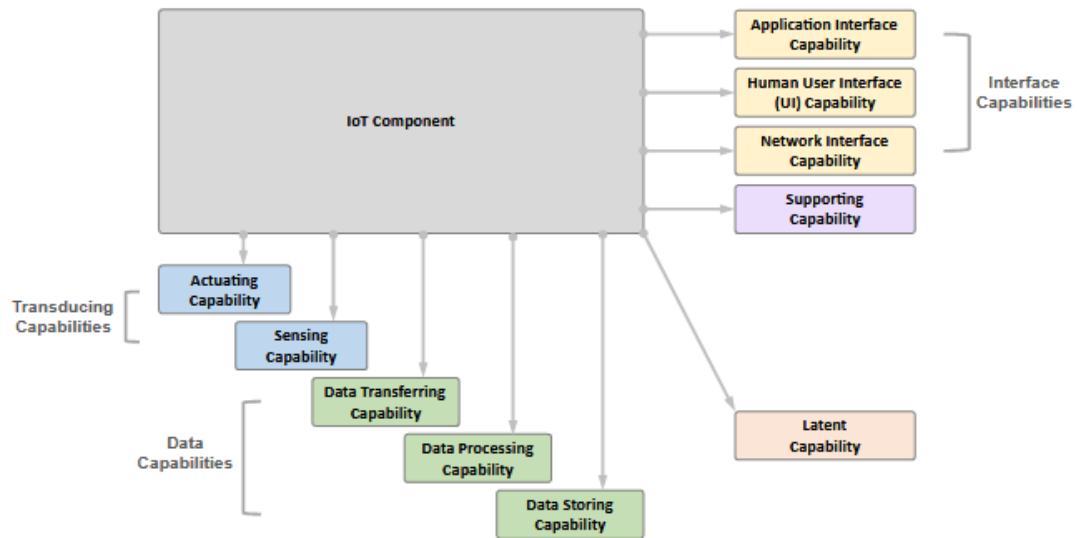


Figure 2.1.: Component Capability Pattern.  
(Young et al., 2022, S. 40)

**Three-Tier Architecture Pattern** The system comprises the Edge, Platform, and Enterprise Tiers, as well as connecting networks. The Edge Tier contains sensors and gateways that collect data. These are connected by the Proximity Network. Data preprocessing may already be happening there.

The Platform Tier is responsible for most data processing and storage via databases. It is connected to the Edge Tier via the Access Network.

The Enterprise Tier provides domain-specific applications and interfaces for end users. These are built upon the processed data from the platform tier. It also issues controls to lower tiers. This tier is connected to the Access Network via the Service Network. The three tiers can also be further divided into different domains. That makes sense for bigger systems. But for a simple system as the one described in this work it is not necessary and therefore these domains will not be explained here.

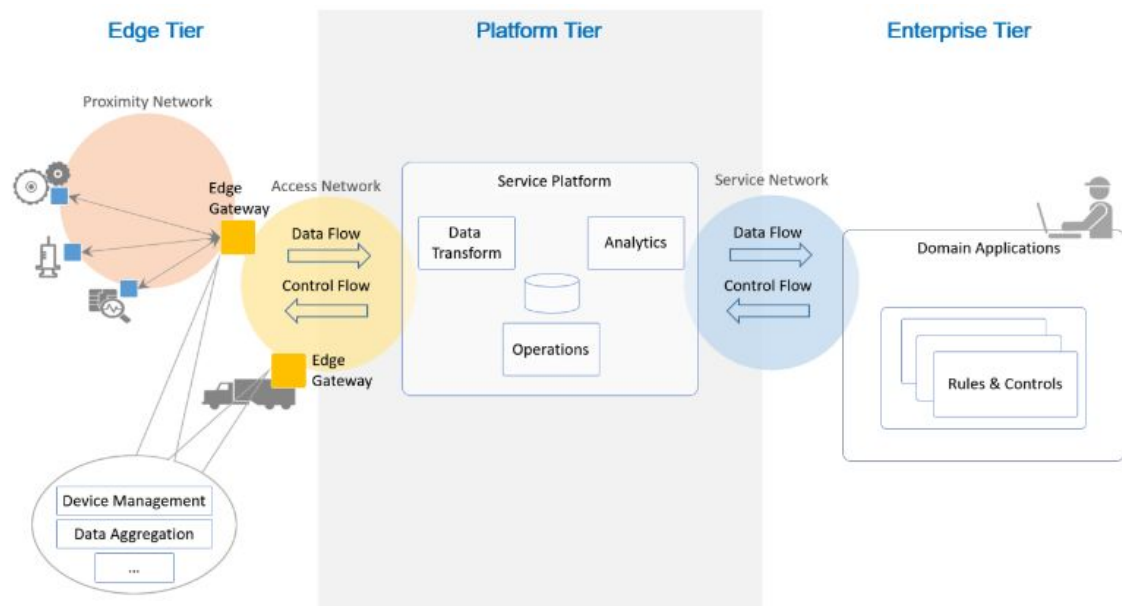


Figure 2.2.: Three Tier Architecture  
(Young et al., 2022, S. 44)

## 2.2.2. Performance measurement in production environments

### 2.2.3. IoT-Platforms

The IoT is known for producing large amounts of data and for the potentials to grow these amounts even more. Therefore a scalable software infrastructure that is needed. That is where IoT-Platforms come into play [Turki et al., 2024]. The authors also mention that IoT-Platforms help accelerating the solution development "[...] by providing foundational capabilities, avoiding the need to implement low-level infrastructure."

[Asemani et al., 2019] further highlight the different capabilities that are typical for IoT-Platforms.

**Connectivity and Device Management** Through various communication protocols the platforms connect with the devices, enabling them to communicate with each other, manage device status and configurations, handle software updates and provide mechanisms for error reporting.

**Data Storage, Management, Analysis, Visualization** Through connections to databases they store large volumes of data often in the cloud or locally. Also further data processing and analytics through various methods as well as visualizations through dashboards are possible.

**Development and Deployment Tools** By providing APIs and SDKs the developers are enabled to further create custom applications.

**Auditing and Payments** The Platforms help to have an overview over the data or compute usage and the resulting costs.

**Service Management** By giving an oversight over parameters like resource consumption, data requirements and access, the user can monitor vertical as well as platform internal services. The platforms also enable the communication between services or combination of basic services to create new ones.



**Integration** Platforms can be integrated with each other, other data sources and the cloud.

**Fog/Edge Computing** IoT-Platforms often support distributed data processing and storage. This can lead to less traffic due to processing close to the data source. Faster transmission would be enabled therefore and reinforced due to shorter communication distances.

The researchers go on to reveal that while commercial platforms carry all of the mentioned capabilities, open source platforms are often focused on specific capabilities thus in implementation sometimes need to be combined to deliver a holistic IoT-Platform.

#### 2.2.4. Differences between Relational and Timeseries Databases

In the paper written by [Türkoğlu et al., 2024] it is analyzed how relational databases and time series databases compare regarding speed and storage efficiency when used in Grafana. First they point out the use case for relational databases is for single time data points, which can be related to other data points over various tables. Therefore enabling complex queries involving joins, aggregations and multiple tables. They make sure the data is accurate and stays consistent. Time series data bases on the other hand are made for data points with a timestamp and large volumes of data. This makes them ideal for IoT applications and real time data analytics. They are optimized to enable high speed read and write operations as well as efficient data storage. The differences in query return time are already there with small amounts of data, but when scaling up the amount they become clearly visible as can be seen in the figure below.

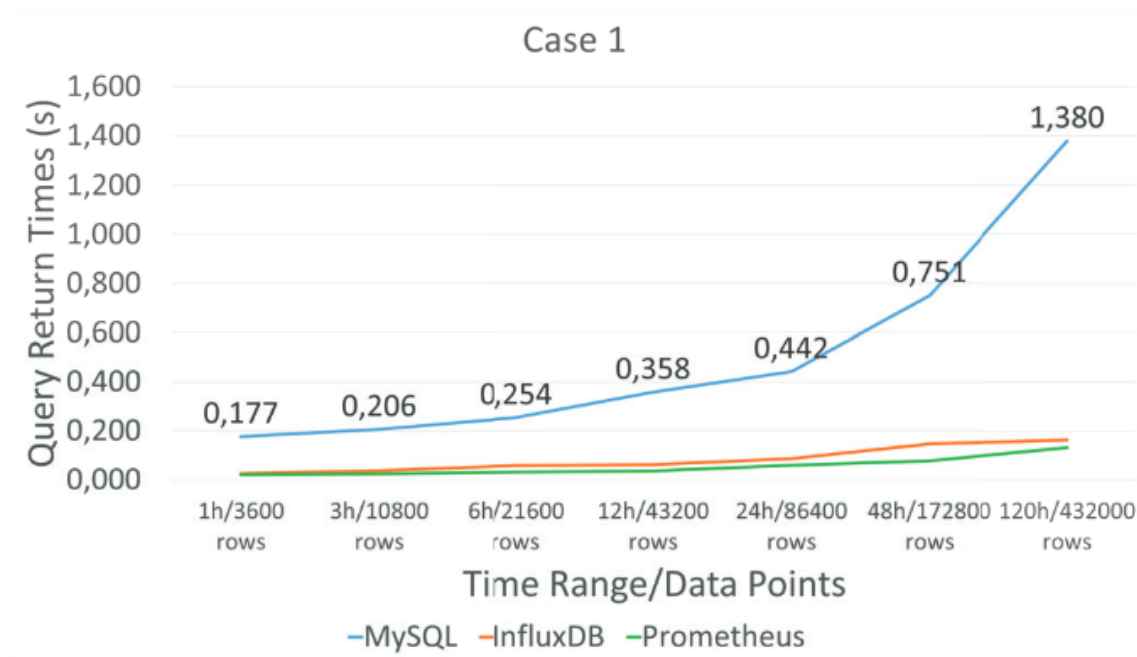


Figure 2.3.: Query Performance Relational vs. Time Series DB  
(Türkoglu et. al, 2024, S. 2)

This is only one of three test cases but they all give a similar picture that shows the superiority of time series databases regarding query return times.

#### 2.2.5. Dashboarding

## **3. Requirements Analysis**

### **3.0.1. Functional Requirements**

### **3.0.2. Non-Functional Requirements**

### **3.0.3. Constraints**

### **3.0.4. KPI Selection and Justification**

The team around [Kang et al., 2016] from the National Institute of Standards and Technologie in the U.S. has worked out the different kinds of KPI's that are being used in operation management and production and how the various metrics and KPI's are related to each other.

First there are the supporting elements which in this thesis will be called metrics because they describe the measured data that is needed to calculate the basic KPI's. The supporting elements are devided into the categories time and quantity.

Then there are maintenance elements which give insight into upkeep of the machines. Of which the following were able to be calculated with the given metrics.

## **4. Zusammenfassung und Ausblick**

# Bibliography

- [Wha, ] What is IoT (Internet of Things)? | Definition from TechTarget.  
<https://www.techtarget.com/iotagenda/definition/Internet-of-Things-IoT>.
- [Asemani et al., 2019] Asemani, M., Abdollahei, F., and Jabbari, F. (2019). Understanding IoT Platforms : Towards a comprehensive definition and main characteristic description. In *2019 5th International Conference on Web Research (ICWR)*, pages 172–177.
- [Kang et al., 2016] Kang, N., Zhao, C., Li, J., and Horst, J. A. (2016). A Hierarchical structure of key performance indicators for operation management and continuous improvement in production systems. *International journal of production research*, 54(21):6333–6350.
- [Rose et al., ] Rose, K., Eldridge, S., and Chapin, L. The Internet of Things: An Overview.
- [Turki et al., 2024] Turki, M., Boussaidi, G. E., Benzarti, I., and Mili, H. (2024). Evaluating Open Source IoT Platforms: A GitHub Analysis. In *2024 IEEE/ACM 6th International Workshop on Software Engineering Research & Practices for the IoT (SERP4IoT)*, pages 14–21.
- [Türkoğlu et al., 2024] Türkoğlu, A., Ersen, O., Yiğit, İ. O., Unal, D., and Golcuk, H. (2024). Comparison Between Time Series and Relational Databases. In *2024 9th International Conference on Computer Science and Engineering (UBMK)*, pages 1174–1177.
- [Young, 2022] Young, D. T.-T. (2022). The Industrial Internet Reference Architecture.

# Abkürzungsverzeichnis

<i>IoT</i>	Internet of Things
<i>IIoT</i>	Industrial Internet of Things
<i>KPI</i>	Key Performance Indicator

# List of Figures

2.1. Component Capability Pattern. (Young et al., 2022, S. 40).....	7
2.2. Three Tier Architecture (Young et al., 2022, S. 44) .....	8
2.3. Query Performance Relational vs. Time Series DB (Turkoglu et. al, 2024, S. 2) ....	9

## List of Tables

## A. Quellcode

1. Source 1
2. Source 2



## **B. Rohdatenvisualisierungen**

1. Graustufen
2. Verteilungen