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 24, 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 24, 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

1. Introduction

Industrial sewing machines are of crucial importance in the textile industry, where they are typically utilized in the final stage of production to assemble the end product. This stage necessitates the highest level of human involvement, thereby becoming a pivotal element in determining both production efficiency and product quality. Therefore, the implementation of performance measurement techniques is particularly appropriate in this context. In the field of performance measurement, the seminal work by Neely [Neely et al., 1995] is widely cited. They defined performance measurement as ”[...] the process of quantifying the efficiency and effectiveness of action.” . This is frequently achieved through the implementation of Key Performance Indicators (KPIs), as they are formally standardized in the ISO 22400 framework, which governs operations management and production.

In recent years, the popularity of automatic systems for performance measurement on sewing machines has increased. Nonetheless, these systems continue to encounter certain challenges that have frequently been overlooked. Firstly, it must be acknowledged that a considerable number of systems are dependent on cloud technology. This reliance engenders certain issues, including elevated latency and the perpetual financial obligations associated with cloud usage. Secondly, the utilization of standards and frameworks is frequently neglected, which results in the complexity of scaling and maintaining these systems. Thirdly, the prevailing focus of numerous works in this field is retrofitting sewing machines, rather than utilizing the machine's inherent data, which often leads to the production of erroneous results. Fourthly, the dearth of software architecture that utilizes services engenders considerable challenges in achieving scalability.

The objective of this thesis is to establish a replicable methodology for designing and implementing a performance measurement system, with a sewing machine serving as a case study. This encompasses the provision of an overview of standards frameworks and technologies, in addition to the demonstration of the selection process for the most suitable option and its subsequent implementation. This thesis proposes a system that maximizes the use of actively maintained open-source technologies while ensuring easy scalability for future expansion. The end result will be a dashboard that provides the most important KPIs (such as cycle time, OEE, setup time, and down time) in real time.

The scope of this work is limited to a Brother sewing machine of type UF-8910, which is connected to a WAGO PLC of type 750-8101 PFC100 CS 2ETH. The WAGO PLC employs the OPC-UA protocol for data transmission to the network. The anticipated data flow and signals are modeled and do not originate from the physical sewing machine and PLC. The sewing machine is part of a shop-floor that is used for workshops where industry customers can gain insight into productivity and quality enhancements within production environments through digitization. The system outlined in this thesis is intended to serve as a demonstrative model, and as such, it will feature visualizations that elucidate its real-time capabilities. The derivation of KPIs must be constrained to those that require querying from the database without necessitating additional post-processing.

The relevance of this thesis is predicated on the increasing demand for data-driven decision-making to optimize efficiency and reduce costs, a phenomenon that is especially pronounced in the highly competitive garment industry. Performance measurement systems empower production management to identify inefficiencies and minimize unproductive periods. Furthermore, they furnish actionable insights that facilitate targeted operator coaching. This thesis makes a significant contribution to the extant knowledge base concerning IoT-based monitoring systems, as it focuses on replicable methods. This as well as the focus on open source technology make the system architecture well suited for small and medium sized companies with limited resources.

This thesis employs a design-science approach, with the sewing machine performance measurement system serving as the artifact. A literature review is also employed to provide a comprehensive overview of the extant related work, as well as the frameworks, standards, and technologies relevant to the subject. To further implement suitable technology solutions, a structured technology selection process is being developed and followed.

In the following, the structure of this thesis is being outlined. The initial section presents the foundational technologies, frameworks, standards, and other groundwork upon which this work is based. The related work section then reviews relevant literature, examining systems with similar objectives to contextualize and position the approach proposed in this thesis. Subsequently, the requirements and system design section details the specific needs addressed by the system, as well as the selected KPIs and technologies. Building on this foundation, the implementation of the system is described. The subsequent evaluation section provides an analysis of the system's strengths and limitations. Finally, the outlook and conclusion offer a summary of the findings and discuss potential directions for future development.

2. Foundations

2.1. Definitions

2.1.1. Takt Time

Maximum time allowed to produce one product in order to meet customer demand

2.1.2. 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 his 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.

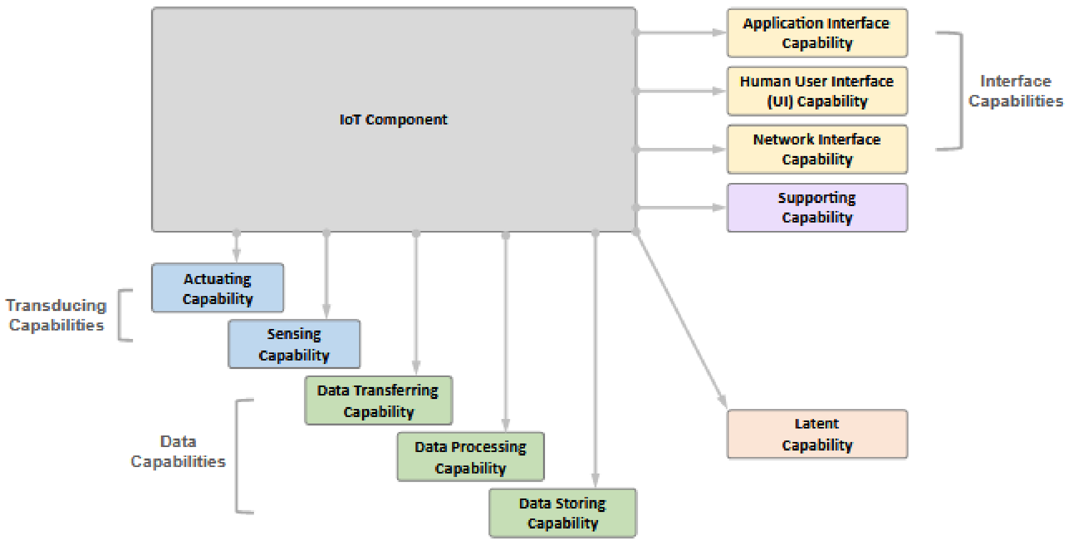

Figure 2.1.: Component Capability Pattern.
(Young et al., 2022, S. 40)

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.

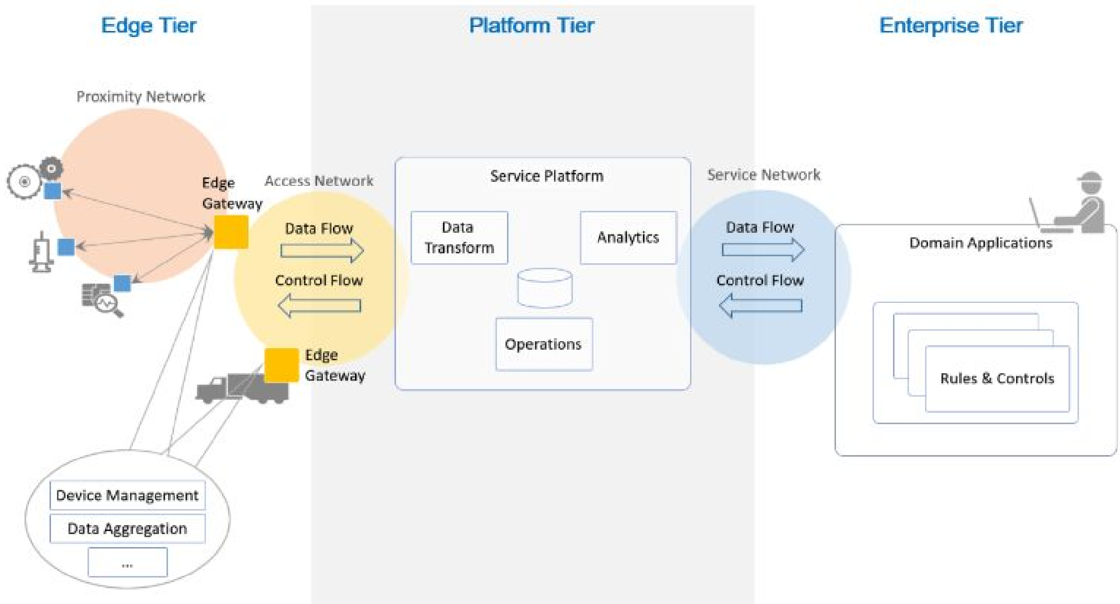

Figure 2.2.: Three Tier Architecture
(Young et al., 2022, S. 44)

2.2.2. KPIs and Metrics for Performance Evaluation in Sewing Operations

Performance measurement in the textile production industry is important due to intense competition. It enables producers to identify potential bottlenecks, provides a deeper understanding of processes, and facilitates more effective resource allocation [Alauddin, 2018].

Key performance indicators (KPIs) and metrics serve as essential tools for performance measurement. Kang et al. [Kang et al., 2016] from the National Institute of Standards and Technology in the United States analyzed the relationships among various types of KPIs and metrics used in operations management and production, based on the ISO 22400 standard. Supporting elements, referred to as metrics in this thesis, describe the measured data necessary for calculating basic KPIs. These supporting elements are categorized into time and quantity. Time elements quantify the duration of various events, such as the production time per unit. Conversely, quantity elements pertain to the number of produced items. Maintenance elements capture information about machine-related issues.

Based on the supporting elements, basic KPIs can be calculated. These KPIs are categorized into production, quality, and maintenance KPIs.

The researchers also emphasize the importance of comprehensive KPIs, which provide a broader overview of production performance. These KPIs build upon basic KPIs and include, for example, Overall Equipment Effectiveness (OEE), which is calculated by multiplying the KPIs for availability, performance, and quality ratio.

Other studies [Kiron, 2022, Alauddin, 2018] have specified KPIs specifically for the sewing section of a textile production plant. In this context, some KPIs overlap with those examined by Kang et al., while additional KPIs unique to the sewing section have also been introduced. The following table classifies these sewing-specific KPIs within the hierarchical framework proposed by Kang et al.

Table 2.1.: Classification of KPIs and Metrics in Sewing Section (based on Kang et al., 2016 and ISO 22400)

KPI/Metric	**Description**	**Classification (Kang et al., 2016 / ISO 22400)**
Supporting Element: Time		
Cycle Time	Total time taken to complete one operation, from start to start of the next piece.	Supporting Element: Time
Standard Minute Value (SMV)	Time required to complete a specific job under standard conditions and pace.	Supporting Element: Time
Allowance	Extra time permitted for personal needs, delays, and fatigue in production.	Supporting Element: Time
Idle Time/Machine	Time when operators or machines are not working, considered lost time.	Supporting Element: Time
Supporting Element: Quantity		
Operation	A step in the process required to convert materials into a finished product.	Supporting Element: Quantity
Manpower to Machine Ratio	Ratio of workers to machines, used to optimize labor and production.	Supporting Element: Quantity
Absenteeism	Rate of operator absence, which affects production and efficiency.	Supporting Element: Quantity
No of Style Change	Frequency of style changes, impacting productivity, efficiency, and quality.	Supporting Element: Quantity
Basic KPI: Production		
Efficiency	Comparison of actual output to what could be achieved with the same resources.	Basic KPI: Production
Productivity	Achievement toward goals based on the relationship between inputs and outputs.	Basic KPI: Production
Availability	Percentage of scheduled time employees or machines are productive.	Basic KPI: Production

KPI/Metric	Description	Classification (Kang et al., 2016 / ISO 22400)
Performance	Amount of product delivered relative to available productive time.	Basic KPI: Production
Line Wise Sewing Efficiency	Efficiency of sewing lines, often linked to man-to-machine ratio.	Basic KPI: Production
Basic KPI: Quality		
Defect per Hundred Units (DHU)	Number of defects found per hundred units produced.	Basic KPI: Quality
Quality	Percentage of perfect or saleable products produced.	Basic KPI: Quality
Comprehensive KPI		
Overall Labor Effectiveness (OLE)	Measures workforce utilization, performance, and quality, reflecting labor's impact on productivity.	Comprehensive KPI
Overall Equipment Effectiveness (OEE)	Quantifies how well equipment performs relative to its designed capacity, considering availability, performance, and quality.	Comprehensive KPI

2.2.3. IoT-Plattforms

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-Plattforms 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.

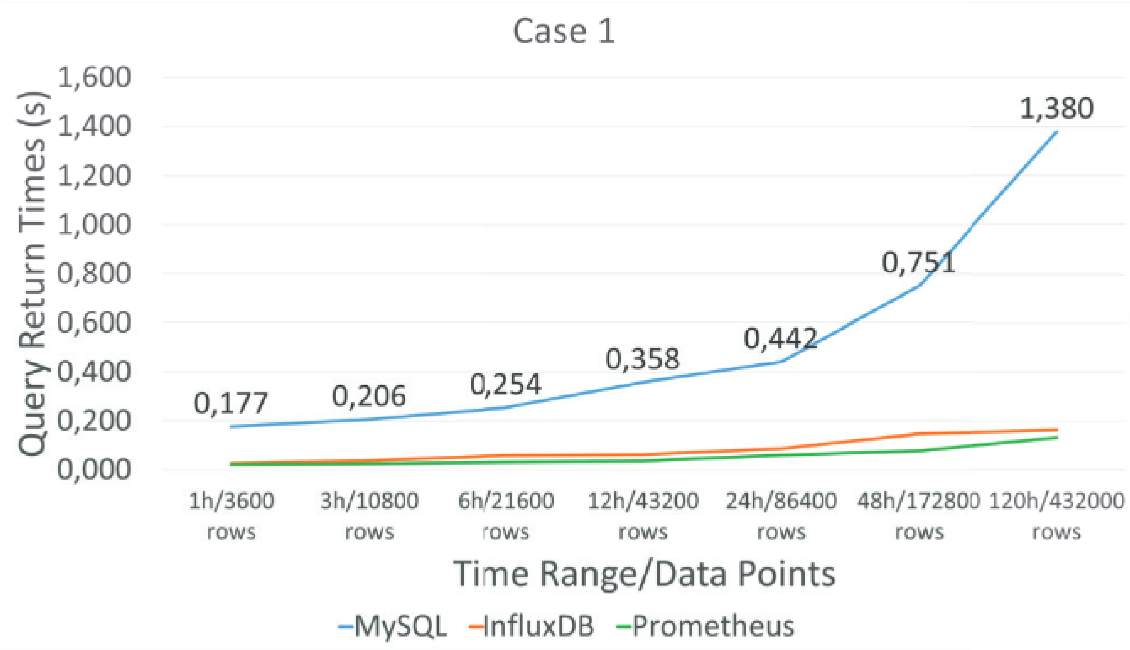

Figure 2.3.: Query Performance Relational vs. Time Series DB
(Turkoglu et. al, 2024, S. 2)

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. Related Work

This section presents an overview of related solutions for sewing machines, with a focus on methods for data collection and processing in monitoring applications. It also examines general approaches to performance measurement in manufacturing. The analysis will likely address the types of metrics underlying these solutions, the sensor technologies used for data acquisition, and the strategies for data processing, storage, and analytics. This review aims to clarify the current state of the art, identify possible gaps in existing systems, and position this work within the field of automated performance measurement systems. Furthermore, the review seeks to highlight potential limitations and opportunities, including technical aspects such as hardware, software, and integration, as well as architectural considerations like standards, frameworks, and open-source approaches, which may influence the implementation of performance measurement systems.

In their seminal work, Jung et al. [Jung et al., 2020] propose a system for the analysis of sewing machine operator skill level and the complexity of the assigned sewing tasks. The present study bears a notable similarity to the aforementioned work, as both analyze the cycle time per unit. This objective is pursued by employing power consumption data of the sewing machines as the fundamental metric. To that end, a power monitoring system is employed, which is connected to the machines power plug without necessitating modifications to the sewing machine itself. Subsequently, the data is transmitted to a cloud server via wireless communication. The determination of quantity and time of work is achieved through the implementation of pattern analysis algorithms. The efficacy of the system is predicated on the non-intrusive nature of PMS. A notable disadvantage of this solution is the necessity of sufficient sample data to ensure the accuracy of the results.
Another system that uses power consumption measurement was proposed by Strazinskas [Stražinskas, 2025]. The utilization of current sensors enabled the discernment of fluctuations in motor current, which are inherently associated with the operational states of the sewing machine, including initiation, cessation, seam length, and stitch speed. The sensor is connected to an Arduino Uno microcontroller, which transmits the received data to a Raspberry Pi 3B+ microcomputer. The microcomputer is responsible for central data collection and storage. The data is stored in files, which is a disadvantage of the system because it is less efficient than storing it in an optimized time series database. Therefore, a greater quantity of space is required, and the rate of data processing is reduced. Additionally, the absence of any preprocessing mechanisms underscores the necessity for optimized data management, a crucial aspect that demands significant memory resources. Strazinskas's research indicates that the utilization of the present sensor engenders measurement errors and noise in the data.
Although adopting a divergent approach, the system proposed by Quoc et al. [Quoc et al., 2025] likewise relies on IoT devices (which were not further described) connected to the machine. The data of these devices is then transmitted to the cloud, where it is integrated with the enterprise's Management Information System (MIS). This integration facilitates the optimization of resource allocation. In light of the aforementioned data, a visualization is generated to illustrate the open positions of a contract and the extent to which it has been completed.
A divergent approach is posited by Wedanage et al. [Wedanage and Thiwanka, 2022] in their study "Fog Assisted Industrial Sewing," in which they utilize a mobile application to record the cycle

times of the sewing step.To initiate and conclude the sewing process on the workpiece, the operator is required to engage a designated button at the commencement and cessation of the operation. The data is structured in a JSON format and published via MQTT under a designated topic. A fog node, implemented using a Cisco IR829 industrial router, is subscribed to the specified topic and processes the data through a Java application. Additionally, the application utilizes a RethinkDB database, which is characterized as open-source and specifically engineered for real-time applications. The fog node is already capable of performing local data analytics. In the cloud, data from all fog nodes is combined to provide real-time alerts and analytics through dashboards. These dashboards visualize cumulative average cycle times per member and compare them to takt time (maximum time allowed to produce one product to meet customer demand). The employment of fog computing facilitates the system's notable scalability, a consequence of the distribution of computing tasks across multiple fog nodes. This configuration facilitates the system's capacity for real-time data analytics and reduced bandwidth utilization. A notable drawback of this system is its reliance on manual input from operators, which introduces a margin of error. Moreover, no historical analytics have been conducted, despite the existence of a foundation for such analyses. A notable drawback of the system in question, as well as those proposed by Jung et al. and Quoc et al., is the utilization of a cloud, which gives rise to several concerns, including but not limited to data security, latency, and ongoing costs. The merits of the aforementioned systems include their capacity for retrofitting older machines and their non-intrusive nature.

Beyond these closely related implementations, a broader perspective is provided by Tambare et al. [Tambare et al., 2022], who review various approaches to performance measurement systems. Initially, the discourse centers on two pivotal standards. The ISA-95 standard delineates entities at the shop floor level, where information technology systems such as ERP, CRM, cloud platforms, and SQL databases interact with operational technology components like sensors, actuators, microcontrollers, SCADA systems, and PLCs. Its primary function is to formalize production processes. Conversely, the ISO 22400 is employed for the formalization of performance metrics. The provided framework facilitates the definition, calculation, and dissemination of key performance indicators (KPIs), encompassing their contextual framework, formula, unit of measurement, range, and intended audience. The researchers then proceed to underscore the Scania case study, wherein these two standards are being utilized, as a paradigm of the practical implementation of international standards for performance measurement in a smart manufacturing context.
This case study by Samir et al. (samirKeyPerformanceIndicators2018) will now be examined in more detail. The system is composed of numerous autonomous, self-contained computational units that are each capable of performing independent computations and collaborating with each other. The software utilized for the processing and distribution of data has been developed using a Service Oriented Architecture (SOA) framework. This approach involves the implementation of multiple services that function independently of one another, with each service designated to specific responsibilities. The communication between the units is facilitated by an Enterprise Service Bus (ESB). The architecture further employs an Event-Driven Architecture (EDA), whereby events serve as triggers for services. This aims to address the issue of communication delays. The KPIs are designed in accordance with the ISO 22400 standard, and the IS-95 is employed to enable automated interfaces between enterprise and control systems.

Taken together, these studies illustrate a variety of approaches to IoT-based machine monitoring, each with specific strengths and limitations. In contrast to the systems examined in the aforementioned studies, the system presented in this thesis is designed to operate with a contemporary sewing machine that facilitates data extraction directly from the machine. It is also intended to provide a more comprehensive overview of the production process by means of various Key Performance Indicators (KPIs), which will be addressed in subsequent sections. The system proposed in this thesis aims to incorporate the strengths of the discussed systems while mitigating their weaknesses. The system has been designed to align more closely with the proposed model in the case study by implementing standards and service-oriented software. Therefore, this facilitates enhanced scalability and maintainability.

4. Requirements Analysis

In order to select the most suitable technologies and design the system, it is first necessary to establish clear requirements. In this chapter, the aforementioned requirements will be enumerated and elucidated. In order to maintain consistency regarding the compliance level, the terms ”must,” ”should,” and ”will” were employed. The following words shall be explained in brief. The term ”must” is employed to signify an unconditional obligation, implying that the fulfillment of this requirement is not subject to discussion or negotiation. The term ”should” conveys a degree of desirability, indicating that fulfillment of the requirement would be advantageous. The term ”will” signifies that this particular requirement is currently under consideration for inclusion in the subsequent release. However, it is imperative to maintain awareness of this requirement so that the system can be designed in a manner that facilitates its seamless integration in the future.

4.0.1. Functional System Requirements

Requirement	**Explanation**
The system must show KPIs that are relevant for the sewing process	In the workshops there must be some KPIs that fit into the story of a textile production with a sewing process
The system should show aditional KPIs that are relevant for the manufacturing industry in general	Workshop participants are from all sorts of companies within the manufacturing industry
The system must present these KPIs in a visual manner that provides information about the classification of the current value e.g. with colors and thresholds	So that the management can act quickly upon the KPIs and does not need to lookup thresholds
The system must provide the user with the ability to change the timeframe on which the KPIs are calculated	Especially when looking at historic data it is usefull to be able to set the timeframe
The system should show graphs with historical data	Enables management to see trends and patterns

4.0.2. Non-Functional Requirements

Requirement	Explanation
The system must make use of open source software where possible	To be replecable by small and medium companies
The system must be capable to generate all of the KPIs from the machine-data without installing any additional sensors	
The system must be designed in a way that makes it easily scalable	Usually more than just one machine need to be considered for monitoring
The system must be deployable with minimal effort	To be able to deploy elsewhere with not much effort
The system must be capable to retrieve raw data via opc-ua	The PLC to which the sewing machine is connected publishes the data over opc-ua
The system should make use of existing patterns, frameworks and solutions were possible	The system shall serve as a reference for other systems that are more readily implementable
The system should update the KPIs in real-time (¡10s)	To support timely interventions
The system must be able to run on a local machine and therefore independent of any cloud service	
The system must provide the user with the ability to access the dashboard from within the shopfloor network	

4.0.3. IoT Platform Evaluation and Selection

The raw data from the PLC had to undergo processing, storage, and visualization. The utility of IoT platforms stems from their ability to execute these functions and frequently extend beyond them. These systems meet the criteria for scalability and leverage existing solutions. To ensure that the selected platform satisfies other requirements and does not conflict with any necessary ones, a decision support framework was developed. The present framework draws inspiration from the one proposed by Gustin and Jasperneite [Gustin and Jasperneite, 2022]. The researchers identified a set of criteria and classified them into categories. The categories encompassed ”Knock-Out,” ”Device Management,” ”Hosting,” ”General,” and ”Effort.” Each category was assigned a weight, with the exception of the ”Knock-Out” category. The ”Knock-Out” category indicates that a criterion tagged with it must be met; otherwise, the platform will not be considered further. The weights are expressed as percentages, and when added together, they equal 100%. In addition, each criterion is assigned a distinct weight, thereby enabling the allocation of greater significance to specific criteria. The evaluation of each platform is subsequently determined by the following formula:

$$s_i = f_i \cdot 100P \cdot weight_{category} \cdot weight_{criterion}$$

In this context, f_i denotes the fulfillment factor. The newly developed framework is distinguished by the following characteristics: The criteria are flexible and should be altered according to the requirements. The aforementioned study concentrates on device management; however, this is not a salient issue in the context of the proposed system. The categories were modified to "Knock-Out," "High Importance," "Mid Importance," and "Low Importance." In this manner, the criteria are not constrained to a particular subject matter; rather, they can be assigned an extent of importance at will. The numerical values assigned to the categories are as follows: 1 for low importance, 2 for medium importance, and 3 for high importance. The new formula for calculating the platforms' score is more straightforward as well:

$$s_i = f_i \cdot weight_{category}$$

The fulfillment factor is typically binary in nature. The value of the variable can be either one or zero; however, in some cases, it can also be fractional. For instance, it would be advantageous for the platforms to support additional communication protocols beyond OPC UA. Specifically, the focus is on HTTP, MQTT, CoAP, and AMQP. In the event that a given platform offers support for only one of these, it is awarded a quarter point. In the event that this is the case, it will be elucidated in the subsequent explanation. The fulfillment of the criteria was determined in the same manner as that employed by the researchers. An investigation was conducted into the documentation, website, and GitHub repository of the platform. In the event that information regarding the availability of the criterion was ascertained, the point was granted. In the absence of pertinent information, the concept in question was deemed to be nonexistent. The subsequent paragraphs will provide detailed explanations for each of the aforementioned criteria. A selection of the materials was obtained from the publication by Gustin and Jasperneite if their utility was deemed to be beneficial. The remaining ones were developed in response to specific requirements.

Knock-Out Criteria

Availability of Documentation
The implementation, deployment, and utilization of the solution are contingent upon the availability of English-language documentation, which is imperative for effective operation. This criterion originates from the seminal contributions of Gustin and Jasperneite
Cloud independence
In order to guarantee that the platform can be hosted on a local server that is accessible via the shopfloor network and that no additional cloud services are required, this criterion was established. This criterion is derived directly from the stipulated requirements.
Ability to process raw data and derive KPI's from it
In order to visualize the KPIs, the data must first undergo processing, after which the KPIs are to be calculated based on the processed data. This criterion pertains to the requirement of calculating KPIs.
OPC-UA capability
The PLC that controls the sewing machine transmits the data via the OPC UA protocol. Therefore, it is imperative that the platform demonstrate its capacity to effectively manage this particular

protocol. This criterion is predicated on the requirements.

High Importance Criteria

Dashboarding Capabilities
In order to facilitate the visualization of KPIs, the platform must possess the capacity to construct dashboards. This criterion pertains to the necessity of possessing the capacity for visualizing KPIs.
Actively Maintained
This criterion is instrumental in ensuring the platform's security and compatibility with evolving technologies and standards.The necessity for this criterion arises from the imperative for open-source software.
Completion of Server SW
This criterion offers insight into the ease with which the software can be installed. A score of one is assigned if a Docker image is provided, and a score of zero is assigned if only source code is available. This criterion is derived from the work of Gustin and Jasperneite
Development of the server-side application
This criterion delineates the ease with which rules can be created and data processed within the application. In the event that a rule engine (low code) is provided, a point will be awarded. In the event that an SDK is provided, a total of 0.5 points will be allotted. This criterion originates from the work of Gustin and Jasperneite
Examples for Server-side implementation
These examples methodically illustrate the server-side implementation process and the anticipated outcomes.This approach has been shown to expedite the implementation process. This criterion originates from the work of Gustin and Jasperneite

Medium Importance Criteria

Supports MQTT, HTTP, CoAP, AMQP
The implementation of these protocols would significantly augment the system's scalability, as it would facilitate the connection of a greater number of machines, gateways, and microcontrollers. This criterion pertains to the necessity of having a scalable system with minimal effort.
Availability of Tutorials
Tutorials are instrumental in facilitating user comprehension of the platform and its applications, thereby accelerating the development process. This criterion originates from the work of Gustin and Jasperneite

Low Importance Criteria

Fault Detection
This feature facilitates the collection of data pertaining to anomalous behavior or fault conditions of the machine. Therefore, it facilitates the identification and resolution of potential issues. While it would certainly be a beneficial addition, this feature does not directly align with the primary objectives of the system. This criterion is derived from the work of Gustin and Jasperneite
Heartbeat Monitor
This feature periodically ascertains the online status of the connected machine. Consequently, this would facilitate the identification of device connectivity issues and ensure system reliability. This feature is commendable, yet it does not directly align with the primary function of the system. This criterion is derived from the work of Gustin and Jasperneite

The platforms to be evaluated were derived from two papers that themselves evaluated IoT platforms. Initially, the open-source platforms from the aforementioned study were selected. These include openBalen, Thingsboard, Kapua, OpenRemote, Ubuntu Core, FIWARE, OpenMTC, Mainflux, and DeviceHive. In a separate study, Turki's 2024 investigation [Turki et al., 2024] focused exclusively on the evaluation of open-source Internet of Things (IoT) platforms, utilizing data derived from GitHub statistics. The following factors were taken into consideration: the stars that were given by users, health, contributors, open issues, closed issues, and releases. Due to the more objective nature of this evaluation, only the top five IoT platforms were selected. The restriction to a specific number was necessitated by the impracticality of evaluating an indefinite number of platforms, as this would have entailed an inordinate amount of time. The top five platforms identified in this study are Thingsboard, Digiot, Mainflux, OpenRemote, and IotSharp. However, it should be noted that only two of these findings were in disjunction with those selected from Gustin and Jasperneite's work. In the course of the research, the Node-RED platform emerged on a regular basis; however, it was not referenced in the evaluated documents. The platform was also mentioned by a colleague, and it was therefore given due consideration. Additionally, UMH was selected for evaluation because it builds upon robust open-source tools, such as Grafana and Apache Kafka, and supports a wide array of protocols.
The ensuing table 4.1 presents the results of the aforementioned calculations. It is imperative to note that only the platforms that met all of the Knock-Out criteria are displayed therein.

Table 4.1.: IoT Platform Evaluation

IoT Platform	**Score**
Thingsboard	19.5
Node-Red	21.5
FIWARE	19.0
OpenRemote	12.0
UMH	20.5

The evaluation metrics for Node-RED, ThingsBoard, and UMH exhibited a high degree of similarity, suggesting comparable levels of suitability among these options. Consequently, an in-depth investigation was conducted to ascertain their system requirements. Of the three options, the UMH configuration exhibits the most demanding specifications. It requires 200 GB of persistent memory, 16 GB of RAM, and an 8-core CPU [UMH,], all of which the server configuration is incapable of meeting. The server configuration, on the other hand, has a specification of 100 GB of persistent memory, 8 GB of RAM, and an 8-core CPU. In the context of Thingsboard, the minimum amount of RAM required is specified as 4 GB [thingsboard,]. According to the official Node-RED website, there are no listed system requirements. However, it has been documented that it is capable of operating on a Raspberry Pi [Run,]. Additionally, certain users have attested to the successful execution of the program on a Raspberry Pi 2 model [Nod, 2020], which possesses a mere 1 GB of RAM and a 900 MHz quad-core central processing unit [Ltd,]. Given its ostensibly minimal complexity and superior performance metrics in the preliminary evaluation, Node-RED was designated as the IoT platform for the system under development.

4.0.4. Database Selection

The selection of a suitable DBMS was imperative for the storage of the data. In the theoretical foundations, the rationale for the superiority of a time series database in the context of IoT streaming data over a relational database was previously delineated. The selection of the Timeseries Database Management System (TS-DBMS) was executed with a lower degree of sophistication compared to that of the IoT Platform. The rationale underlying this matter is predicated upon the temporal limitations imposed on the undertaking. A comprehensive investigation was conducted to ascertain the most prominent TS-DBMSs on db-engines.com, a website dedicated to determining the popularity of DBMSs. The following metrics have been identified as contributing factors to the popularity ranking:

- The number of mentions on websites
- General intrest in the system using Google Trends
- The number of related questions and posts on well-known IT Q&A sites
- The number of job offers in which the system was mentioned
- The number of mentions on professional networks
- The number of mentions on social networks

Subsequently, the capacity of InfluxDB to process boolean values was ascertained. This assertion was corroborated in the documentation [Wri,]. In addition, an assessment was conducted to ascertain whether InfluxDB is compatible with Node-RED. The package that was found (node-red-contrib-influxdb 0.7.0) has been determined to only support InfluxDB 1.x and 2.0 [Nod,]. Prior to this, it was determined that InfluxDB 3 Core's capacity for querying data is confined to a span of merely 72 hours [Que,], a limitation that was deemed inadequate in view of the necessity to compute historical KPIs. Consequently, InfluxdDB v2 was selected due to its compatibility with Node-RED and its capacity to support unlimited querying windows.

4.0.5. Dashboarding Tool Selection

Initially, the objective was to leverage the dashboarding capabilities of the selected IoT Platform Node-RED. However, it appears that the utilization of this particular node-RED feature for visualization purposes is not a prevalent practice. In contrast, another software program, Grafana, was utilized in a significant number of cases similar to the one under consideration in this study. Consequently, the decision was made to utilize Grafana for the development of dashboards, with the objective of facilitating the visualization of the KPIs. This approach was adopted to enhance the probability that other colleagues would be acquainted with the solution, thereby facilitating a more streamlined handover process. The decision was further corroborated by Rani in his work [**?**], who states that it is highly appropriate for time series visualization and monitoring. Additionally, the integration of the system with various databases, including InfluxDB, is emphasized.

4.0.6. System Architecture

The system architecture was designed in accordance with the three-tier architecture pattern from the IIRA mentioned in the foundation chapter. The Three Tier Architecture Pattern was selected because it exhibits a clear separation of concerns between the edge, platform, and enterprise tiers, thereby facilitating enhanced maintainability. Additionally, the progression of the distinct components is rendered more straightforward and less susceptible to errors in comparison to a monolithic application. The selected architecture pattern offers a distinct advantage, namely the seamless integration of supplementary devices and services. The incorporation of devices into the Node-RED system is contingent upon their connectivity to the shopfloor network. The integration of services at the platform or enterprise tier can be accomplished without the need for significant alterations to the existing system, thereby facilitating flexible expansion and adaptation to novel requirements.

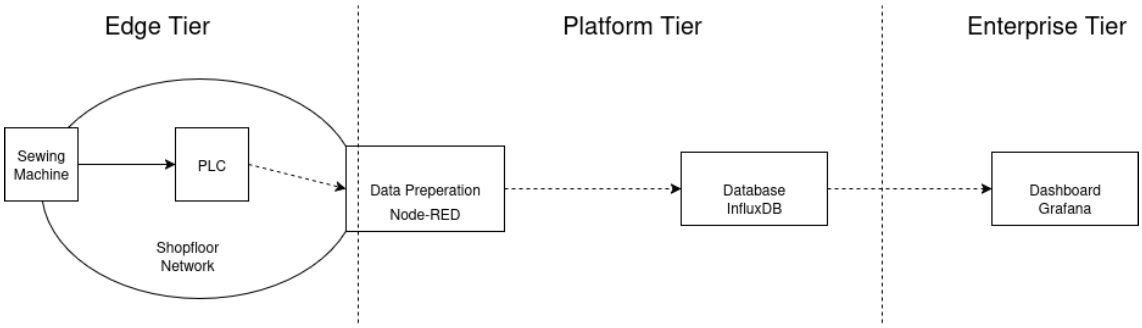

Figure 4.1.: System Architecture

The signal outputs from the sewing machines are wired to the PLC, as illustrated by the continuous line. Conversely, the dashed lines signify the transmission of data either over a network or within a server. The PLC provides these signals via OPC UA, and it is accessible via the shopfloor network. In the context of this architecture, Node-RED performs a variety of functions. Initially, it functions as an IoT gateway, thereby aggregating data from disparate sources. In this particular instance, the sole source of data is from the PLC. However, the component is also capable of performing data processing operations. Subsequently, the data is transmitted to the database. From there, it can be queried with the dashboarding solution.

4.0.7. KPI Selection and Justification

Of which the following were able to be calculated with the given metrics.

5. Zusammenfassung und Ausblick

Bibliography

[Nod,] Node-red-contrib-influxdb. http://flows.nodered.org/node/node-red-contrib-influxdb.

[Que,] Query data | Get started with InfluxDB 3 Core | InfluxDB 3 Core Documentation. https://docs.influxdata.com/influxdb3/core/get-started/query/#Copyright.

[Run,] Running Node-RED locally : Node-RED. https://nodered.org/docs/getting-started/local.

[UMH,] UMH installation requirements. https://umh.docs.umh.app/docs/getstarted/installation/.

[Wha,] What is IoT (Internet of Things)? | Definition from TechTarget. https://www.techtarget.com/iotagenda/definition/Internet-of-Things-IoT.

[Wri,] Write data to InfluxDB 3 Core | InfluxDB 3 Core Documentation. https://docs.influxdata.com/influxdb3/core/write-data/#line-protocol-elements.

[Nod, 2020] (2020). Node red minimun configuration - Hardware. https://discourse.nodered.org/t/node-red-minimun-configuration/31416/2.

[Alauddin, 2018] Alauddin, M. (2018). Process improvement in sewing section of a garments factory a case study.

[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.

[Gustin and Jasperneite, 2022] Gustin, D. and Jasperneite, J. (2022). IoT Device Management Based on Open Source Platforms - Requirements Analysis and Evaluation. In *NOMS 2022-2022 IEEE/IFIP Network Operations and Management Symposium*, pages 1–4.

[Jung et al., 2020] Jung, W.-K., Park, Y.-C., Lee, J.-W., and Suh, E. S. (2020). Remote Sensing of Sewing Work Levels Using a Power Monitoring System. *Applied Sciences*, 10(9):3104.

[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.

[Kiron, 2022] Kiron, M. I. (2022). KPI (Key Performance Indicator) in Garment Industry.

[Ltd,] Ltd, R. P. Buy a Raspberry Pi 2 Model B. https://www.raspberrypi.com/products/raspberry-pi-2-model-b/.

[Neely et al., 1995] Neely, A., Gregory, M., and Platts, K. (1995). Performance measurement system design: A literature review andresearch agenda. *International Journal of Operations & Production Management*, 15(4):80–116.

[Quoc et al., 2025] Quoc, H. D., Thuy, T. T. T., Sinh, Q. N., Danh, H. B., and Lan, A. V. T. (2025). Applying IoT for Operations Management and Production Planning in Industrial Sewing. In Nghia, P. T., Thai, V. D., Thuy, N. T., Huynh, V.-N., and Van Huan, N., editors, *Advances in Information and Communication Technology*, pages 1054–1062, Cham. Springer Nature Switzerland.

[Rose et al.,] Rose, K., Eldridge, S., and Chapin, L. The Internet of Things: An Overview.

[Stražinskas, 2025] Stražinskas, P. (2025). *Development of a Solution for Monitoring and Recording the State of a Sewing Machine /.* PhD thesis, Kauno technologijos universitetas.

[Tambare et al., 2022] Tambare, P., Meshram, C., Lee, C.-C., Ramteke, R. J., and Imoize, A. L. (2022). Performance Measurement System and Quality Management in Data-Driven Industry 4.0: A Review. *Sensors*, 22(1):224.

[thingsboard,] thingsboard. Installing ThingsBoard CE on Ubuntu Server. https://thingsboard.io/docs/user-guide/install/ubuntu/.

[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.

[Wedanage and Thiwanka, 2022] Wedanage, D. K. H. W. and Thiwanka, G. M. (2022). Fog Assisted Industrial IoT Module for Cycle Time Capturing in Apparel Industry. In *2022 International Mobile and Embedded Technology Conference (MECON)*, pages 352–356.

[Young, 2022] Young, D. T.-T. (2022). The Industrial Internet Reference Architecture.

Abkürzungsverzeichnis

IoT	Internet of Things
$IIoT$	Industrial Internet of Things
KPI	Key Performance Indicator
PLC	Programmable Logic Controller
$OPCUA$	Open Process Control Unified Architecture
$DBMS$	Database Management System
$TS-DBMS$	Timeseries Database Management System
$IIRA$	Industrial Internet Reference Architecture
x	x

List of Figures

List of Tables

A. Quellcode

1. Source 1
2. Source 2

B. Rohdatenvisualisierungen

1. Graustufen
2. Verteilungen