DISTRIBUTED MQTT BROKER ARCHITECTURE

AMBIENT INTELLIGENCE RESEARCH GROUP

ZIBUSISO MASUKU
APPLIED COMPUTER SCIENCE STUDENT
Saxion University of Applied Sciences

Place, date:	Enschede, Netherlands		
Drawn by:	Ambient Intelligence Research Group Intern		
	Zibusiso Masuku	509483	509483@student.saxion.nl | z.t.masuku@saxion.nl
Version number:	[0.1]: *Initial Project Plan with content filled in*		
Clients:	Ambient Intelligence Research Group		

Table of Contents

List of Figures

List of Tables

Abbreviations

Term	Abbreviation
Ambient Intelligence Research Group	AMI
Denial-of-Service	DoS
Internet of Things	IoT
Message Queuing Telemetry Transport	MQTT
Must Have, Should Have, Could Have, Won't Have	MoSCoW
Service Level Expectation	SLE
Single Point of Failure	SPOF
Weighted Shortest Job First	WSJF

Introduction

This **project plan** details the research assignment that will be carried out by the interning researcher, **Zibusiso Masuku**, an Applied Computer Science student, as part of a **graduation internship assignment** from **February 2025** to **June 2025**. This research assignment will be conducted under and supervised by the **Saxion Ambient Intelligence Group** at **Saxion University of Applied Sciences**. The research motivation aligns with the group's focus in connected embedded systems, data-driven intelligence, and **Internet of Things** (IoT) communication. This section provides an overview of the organization offering the research assignment and the relevance of the project topic to the organization's goals. There is also an explanation on the significance of **MQTT** in IoT systems, its limitations in a centralised architecture, and the possible benefits of a distributed approach. This background will help establish the context for the research and implementation.

Ambient Intelligence Research Group

Figure 1: Ambient Research Group logo

The **Ambient Intelligence Research Group (AMI)** at Saxion University specializes in **sensor technology, artificial intelligence, and smart systems**. The group's mission is to develop IT solutions that enhance connectivity, efficiency, and intelligence in various domains, including **smart industry, sustainable energy, and real-time monitoring systems** [1]. A key research area within AMI is **connected embedded systems**, which focuses on **data distribution, communication protocols, and software development for IoT**.

The **distributed MQTT broker research** aligns perfectly with this focus, as it explores how **real-time sensor data can be effectively transmitted in large-scale, high-traffic environments** without centralized MQTT broker constraints. **AMI**, now referred to as **the organization**, aims to address all possible concerns that come with operation using a centralized MQTT broker with this research assignment and provide an ideal foundation for experimenting with the distributed MQTT models or architectures that will be explored during this research internship.

MQTT and IoT Communication

The Internet of Things (IoT) is transforming modern computing by integrating physical and digital environments, enabling **real-time monitoring** and **automation,** allowing for **data-driven decision-making** [2]. IoT systems rely on a vast network of **sensors,**

actuators, and gateways to collect, process, and transmit critical data across connected devices. Ensuring **secure, reliable, and efficient communication** between these devices is fundamental to IoT deployments [3]. The **Message Queuing Telemetry Transport (MQTT)** protocol has been a key communication standard, specifically designed to support **low-bandwidth, high-latency, and resource-constrained environments**. MQTT follows a publish-subscribe (pub-sub) architecture, where clients (IoT devices) communicate indirectly via a broker that manages message distribution [4]. Traditional MQTT implementations rely on a single centralized broker that handles communication of one or more devices that are either a publisher or a subscriber. A publisher is any device that sends a message on a specific topic [5] which is a label on the message used by a broker to differentiate between messages [4]. A subscriber device then subscribes to one or multiple topics and will then receive all messages sent under those topics from the broker.

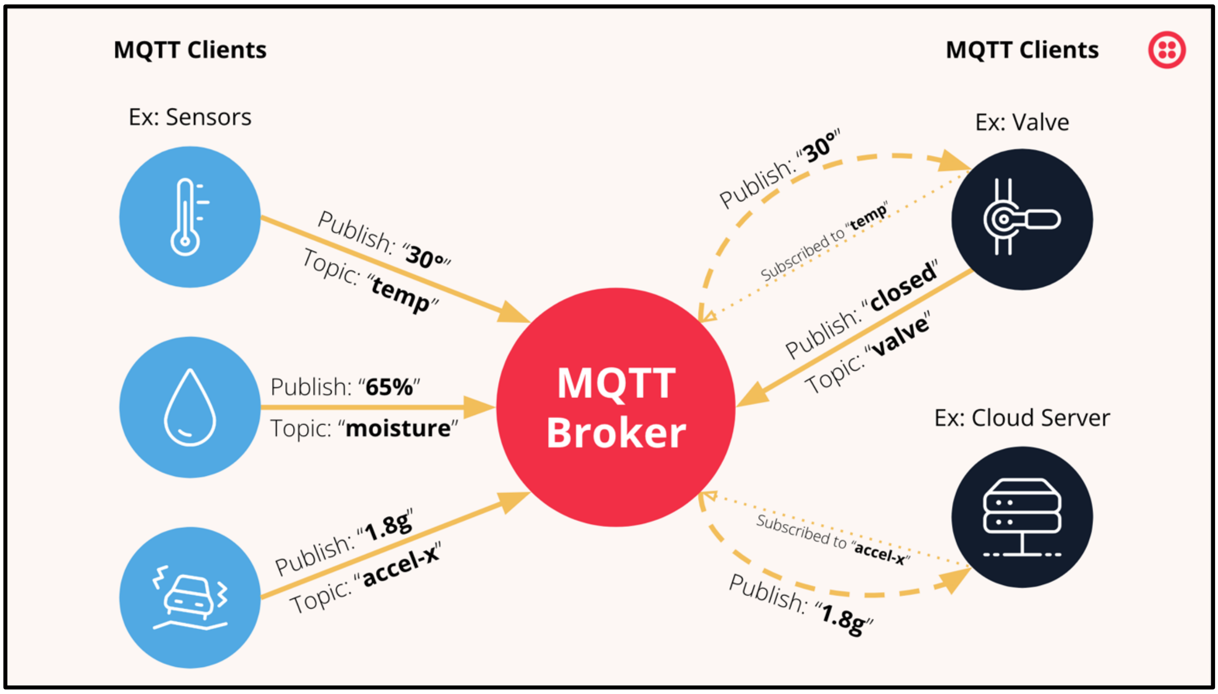


Figure 2: MQTT Broker Illustration

Role of MQTT Brokers

Acting as an **intermediary between publishers and subscribers**, the broker is responsible for **receiving, filtering, and distributing messages** to the appropriate clients based on predefined topics [6]. This eliminates the need for direct device-to-device communication, simplifying network management and ensuring efficient data transmission in resource-constrained environments. In addition to facilitating data exchange, the MQTT broker plays a critical role in managing network scalability and resource optimization. By handling multiple concurrent connections, the broker efficiently distributes messages, reducing bandwidth consumption and processing overhead on individual devices [7]. This is beneficial in environments where IoT systems such as industrial automation and environmental monitoring systems with hundreds or even thousands of sensors all sending data at high frequencies. Security is another crucial aspect of MQTT brokers, as they **control access to communication channels** and enforce authentication mechanisms.

Purpose of Assignment

Centralized brokers present inherent vulnerabilities, including **single points of failure and potential attack vectors such as denial-of-service (DoS) attacks and unauthorized access.** Additionally, the traditional centralized MQTT broker model faces significant challenges in large-scale IoT ecosystems. As the number of connected devices increases, a single broker can become a bottleneck, leading to latency issues, reduced message throughput, and system failures during high-traffic periods [6]. This limitation directly impacts real-time applications where **delays in data transmission** can **lead to system inefficiencies** and **potential failures**. To address these issues, **distributed MQTT architectures** have been proposed as a **scalable alternative**, allowing message routing and processing to be distributed across multiple brokers. This approach improves **load balancing, fault tolerance, overall system efficiency and eliminates other various failures faced by the traditional implementation like network attacks** [8], ensuring that IoT networks can **accommodate increasing data volumes without compromising performance**.
This research assignment aims to provide **practical insights into scalable IoT communication frameworks,** by evaluating the **performance metrics such as latency, message throughput, and fault tolerance** of the different architectures currently in the market and potentially improving on the solutions. The findings will contribute to **enhancing IoT network efficiency, reducing communication delays, and supporting the seamless expansion of interconnected systems**.

Methodology: Agile and Kanban

Agile Methodology

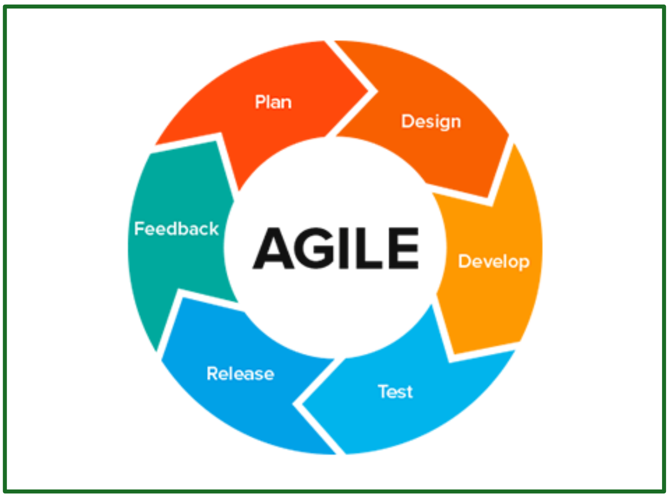


Figure 3: Agile Methodology

Agile methodologies prioritize iterative progress, flexibility, and responsiveness to change. Due to the research-oriented nature of this internship, an Agile approach will be fitting and would ensure flexibility, continuous improvement, and iterative refinement of the developed solutions. Unlike other methodologies, Agile does not enforce a strict sequence of phases or steps but encourages incremental development and continuous feedback loop. The methodology can be use with other methodologies like Scrum, which employs the use of daily standups and fixed iterations or sprints. However, for this assignment Agile will be implemented using Kanban, a visual workflow management system which tracks tasks in different stages of completion.

Kanban Methodology

Kanban allows for **continuous workflow adjustments**, making it particularly suitable for a **research-oriented** work where tasks may evolve based on findings. The workflow follows a logical progression, starting from task definition **product backlog** columns like “Project Plan” which hold the **tasks written as user stories** or in the perspective of the end user of the result. The user stories then **move through the various stages** like “In Progress”, “Review”, “Testing” **until it reaches “Done”** at the end. This ensures that tasks and **solutions are refined and validated before completion** resulting in a finished product that will also be **approved by stakeholders** as part of the review process. Additionally, unlike fixed methodologies such as the V-Model, where phases must be completed sequentially, Kanban allows for **dynamic task prioritization** meaning **tasks can be moved back** to previous phases if needed.

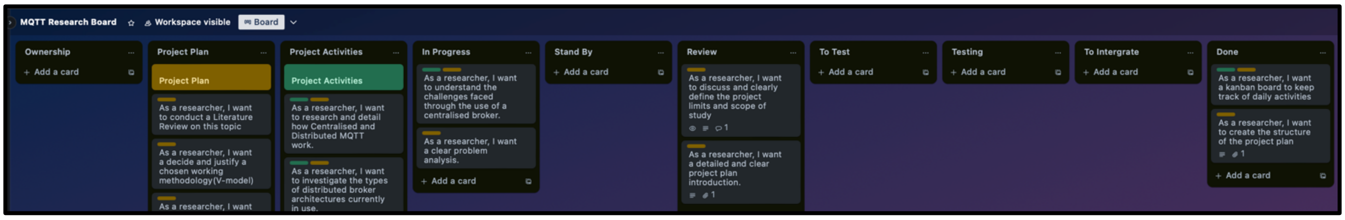


Figure 4: Kanban board for assignment

Daily standup meetings will also be **replaced by weekly check ins, and Bi-Weekly progress meetings with the internship supervisor** to provide updates and receive **feedback**, ensuring continuous improvement while allowing for **independent work**. To ensure that tasks are completed within a reasonable timeframe, **Service Level Expectations (SLEs)** are defined for different categories of tasks. The **SLEs act as a guide for how long the tasks should take** and are not a hard deadline, providing flexibility while maintaining accountability. However, **if a task is finished** quicker than estimated **the next prioritised tasks in the product backlog can be pick up** and moved to in progress.

Report Outline

This document is the project plan for the research assignment offered by the organization, which aims to find the best solution for a distributed MQTT architecture. The document starts with an **Introduction** that establishes research context, motivation and research objectives while justifying the need and relevance of the research topic. There is also an overview of the organization’s goals and interest in the research project. The **Methodology** section describes the Agile methodology and Kanban workflow that will be used during research process. It will also justify the use of Kanban work flow over other Agile methodologies. The **Problem Analysis and Project Objectives** section outlines the limitations of centralized MQTT architectures,and introduces the primary research goal and the sub-research questions guiding the project. The **Project Boundaries** section defines the scope and limitations of the research, clarifying what aspects will be explored and which will be excluded.

The **Project Activities** section details the tasks that will be executed throughout the project timeline. It also explains how activities are prioritized using a Weighted Shortest Job First (WSJF) model. The **Products** section then outlines the tangible deliverables that will be expected at the end of the project, differentiated between software, hardware, and documentation. The **Quality Control** section describes the measures or tools that will be used to ensure the accuracy and reliability of the project outcomes. The **Project Organization** Section outlines the reporting structure of the organization, and shows the researchers role in that structure. The **Planning** section details the project timeline, highlighting the key phases, milestones, and deadlines, ensuring alignment and avoiding delays due to dependacies. The **Costs and Benefits** section addresses costs related to hardware, software, network infrastructure, and time commitments, alongside the expected benefits that the research will bring. Finally the **Risk Analysis** section examines the potential project risks, including technical challenges, time constraints, and implementation complexities.Mitigation measures are detailed if possible and acceptable risk are also noted.

Problem Analysis and Project Objectives

This section defines the key challenges associated with **centralized MQTT architectures** and establishes the need for a **distributed approach**. It examines critical issues such as **scalability limitations, reliability concerns, and performance bottlenecks**, highlighting how current implementations **fail to meet the demands of large-scale IoT systems**. The primary research goal is also clearly defined in this section along with sub-research questions that will help archive the primary goal that will be explored throughout the research assignment.

Challenges of Centralized MQTT Brokers

While the traditional centralized architecture provides a **simple and effective communication model** for small to medium-scale deployments, a primary limitation arises due to its **reliance on a single broker**. This leaves the broker susceptible to various challenges like **bottlenecks, scalability constraints, single points of failure, latency issues, and security vulnerabilities**.

Bottlenecks in Centralized MQTT Systems

In **centralized MQTT architectures**, the broker is responsible for handling all **message routing, storage, and distribution** between publishers and subscribers. As the number of connected devices increases, the broker's processing capacity can become overwhelmed, creating a **communication bottleneck** that leads to **delays** [9]**, message loss, and degraded system performance** due to excessive demand on system resources. Addressing these bottlenecks requires **distributing the workload across multiple brokers**, ensuring that no single point in the system becomes overwhelmed [8]. A **distributed MQTT architecture** can effectively mitigate these issues by allowing messages to be processed in parallel, improving **scalability, fault tolerance, and overall system reliability**.

Scalability Challenges

One of the primary limitations of centralized MQTT systems is the **fixed processing capacity of the broker**. Since all messages pass through a single broker, its ability to handle growing traffic is constrained by available **CPU power, memory, and network bandwidth** [10]. This then limits on the number of simultaneous connections and leads to longer message queues causing delays, packet loss or complete system failure. Additionally, geographical scalability is a challenge in centralized MQTT systems. By **spreading the workload across multiple interconnected brokers**, message processing and network load can be efficiently managed, ensuring **lower latency, higher availability, and improved resilience** in large-scale IoT systems [10].

Single Points of Failure

Single point of failure (SPOF) refers to a **critical component in a system whose failure can cause the entire system to malfunction or become inoperable**. This is common in a centralized MQTT broker system as communication will fail if the broker is unreachable due to software, hardware or network reasons resulting in service disruption. In a **distributed system**, multiple brokers work together to distribute the workload, ensuring that if one broker fails, others can take over seamlessly, maintaining uninterrupted service [11].

Latency and Throughput

Latency refers to the **time delay between a message being sent and received**, while **throughput** measures the **rate at which messages are processed and delivered within a given timeframe**. In a centralized broker architecture, high traffic rates lead to message queuing and processing delays. This is the primary cause for high latency and low processing rates or higher throughput in this type of architecture.
Additionally, **geographical distance** plays a role in increasing latency. When IoT devices are distributed across multiple locations but rely on a **single, remotely located broker**, the physical distance between client nodes and the broker introduces **network propagation delays**. This can be solved by spreading the message handling across multiple brokers, in different locations across the network area, to ensure that client devices are closer and have equal access to the brokers. Additional mechanisms can be implemented to allow devices to choose a desired broker with the lowest latency and throughput, helping with system performance and load balancing.

Security

In a centralized MQTT broker system, an attacker who gains access the to the brokers can intercept or alter messages or halt system functioning and compromising the systems reliability and data integrity. Traditional MQTT communications standards tend to neglect security measures which can leave system vulnerable due to lack of authentication and lack of access control functions [12]. Additionally, cyber-attacks such as **Denial-of-Service (DoS) attacks**, where malicious actors intentionally **overload the broker with excessive connection requests or message traffic**, can be a concern in centralized systems as the broker will not be able to handle the load leading to all the challenges discussed above. Distributed MQTT architectures, in contrast, can **mitigate the impact of these attacks by balancing traffic across multiple brokers**, reducing the risk of a single point of failure.

Research Goal

*The primary research goal involves **research into existing distributed MQTT broker architectures, design and implementation** of some of the desired architectures while **evaluating the performances**, to ensure they meet the need of **enhancing scalability, fault tolerance, performance, and security** in large-scale MQTT broker systems.*

Sub-Research Questions

1. How is the performance of a centralized MQTT broker system and what are its limitations?
2. What are the fundamental designs currently developed that archive an effective distributed MQTT broker system and how do they differ in terms of message routing, load balancing, and fault tolerance??
3. What scalability strategies can be integrated into a distributed MQTT architecture to efficiently handle high traffic loads and a growing number of connected devices?

4. How can session persistence and state synchronization be managed across multiple brokers in a distributed system?
5. What performance benchmarks and metrics should be used to evaluate the effectiveness of a distributed MQTT broker architecture?
6. What security considerations must be accounted for when deploying a distributed MQTT broker system?

Project Activities

This section details the specific tasks that will be performed during the project. The activities include researching the different types of distributed MQTT broker architectures, implementing some of the desired solutions, testing performance, benchmarking the performances against each other, and documenting findings.

Product Backlog

The product backlog listed below are the key tasks that will be conducted through the research internship. Each backlog item is **graded using Weighted Shortest Job First (WSJF) prioritization**, which considers multiple factors to determine the optimal order for task execution. Each factor is rated on a **scale of 1 to 10**, with **higher values indicating greater importance**. The **weights** assigned to each factor are:

- **Project Value (40%)** – How essential the task is to achieving the research goals.
- **Risk (30%)** – The extent to which the task faces risks or dependencies.
- **Time Sensitivity (20%)** – The urgency of the task in relation to deadlines or dependencies.
- **Estimated Effort (10%)** – The difficulty or complexity of completing the task.

Table 1: WSJF List

Backlog Item	Category	Project Value (40%)	Risk (30%)	Time Sensitivity (20%)	Effort (10%)	WSJF Score
Develop a Project Plan	Documentation & Deliverables	10	9	10	5	9,2
Implement, Test and evaluate 3-5 different distributed architectures	Implementation	9	10	7	9	8,9
Investigate implemented distributed MQTT architectures	Research	9	8	9	5	8,3
Literature research/review	Planning	8	8	9	7	8,1
Write Technical Document	Documentation	10	7	5	6	7,7
Implement fault tolerance mechanism on the developed solutions	Implementation	9	6	7	6	7,4

Research and implement security safeguards on developed solution	Implementation	8	6	6	7	6,9
Investigate and analyse broker synchronisation techniques	Analyse	7	8	5	4	6,6
Investigate and analyse the best load balancing strategies for distributed MQTT brokers	Analyse	7	7	5	4	6,3
Research message routing techniques	Research	8	5	5	5	6,2
Present finding to stakeholders	Meetings	8	3	7	4	5,9
Perform testing and bench marking on the developed solution	Testing	8	4	5	5	5,9
Implement basic centralized broker	Implementation	6	6	4	6	5,6
Write Learning Report	Documentation	7	3	4	4	4,9

This list is not fixed and can change. New tasks can be added or removed, and smaller tasks picked up and done concurrently with other larger tasks if they do not affect or block other tasks.

Project Boundaries

This section defines the boundaries of the project, including its focus areas and constraints. It clarifies what is included and excluded in the research and implementation phases.

Project Duration

This research assignment will be carried out over 20 weeks from February 4th, 2025, to June 22nd, 2025. Deadlines and submissions throughout the project have been outline in the Gantt chat in “Appendix A: Gantt Chart”

Project Scope: MoSCoW Method

MoSCoW is an acronym that stands for “Must Haves”, “Should Haves”, “Could Haves” and “Won’t Haves”. It is a popular prioritization method in which requirements are labelled and therefore easier to prioritize.

Must Haves
These are **non-negotiable features** that are essential for project completion. These features are necessary for users to complete tasks with your product. They are also required to meet project goals and expectations set by the client. Most of the development time and resources go to the Must Haves

Should Haves
This involves features that are **important but not as crucial** as the must haves. These features if not implemented can cause risks to the overall success of the final product, but the risk is lower than that of the must haves.

Could Haves
The could haves are **not important to the core function of the product but enhance** the use of the product significantly or do nothing at all. The could haves are not a risk or failure but can be addressed if feasible. They are usually treated like a sprint filler when other important features have already been realized.

Won’t Haves
These features are usually **set aside for the end** of the project timeline if time is available or **for future development** of the product after the end of the current project as improvements. They are not required to pass or part of the “definition done” but a desirable suggested by the client or end users to improve the product even more.

Table 2: MoSCoW Analysis

Must Haves	Should Haves	Could Haves	Won’t Haves
Design and implement at least 3 to 5 solutions for the	Multiple client devices connected and	Cloud based distribution architecture	Integration with third party IoT platforms

distributed broker architecture.	communicating on the network.		
Bench marking mechanism and monitoring tools	Handle varying levels of traffic load	Automated broker discovery and connection	Artificial Intelligence/ Machine learning-based broker load optimization
Implemented Broker synchronisation	Handle simulated points of failure tests and broker switching.	Simulation of large-scale real-world IoT deployment scenarios	Production level/ready distributed MQTT system
Implemented Load balancing across the multiple brokers	Automated device/client scaling mechanism	Visualization tools for monitoring distributed broker traffic	
Implementation of Authentication and encryption measures	Message deduplication and advanced error handling across brokers		
Documentation and Source code or otherwise for the developed solutions			

Products

This section outlines the expected deliverables of the research project. These will include **implemented proof-of-concept MQTT broker architectures**, **performance evaluation tools**, and **documentation** about the implementation details, and recommendations. The deliverables are grouped into the **three sub-categories: software, hardware, and documentation.**

Software

Proof of Concept: Distributed Broker Architecture

The objective of this assignment is to investigate the most effective distributted MQTT broker architectures that solve the challenges associated with the centralized MQTT broker systems. Developing a fully functional, industry-ready solution is not a requirement. Instead, the researcher is expected to provide **well-supported recommendations** for the most suitable architecture, backed by a **working proof of concept** that demonstrates its functionality. To ensure that multiple approaches are explored, the researcher will present at least **two additional alternative architectures alongside the recommended solution**. A **comparative analysis** will be conducted, providing **justification for the final selection** based on performance, scalability, fault tolerance, and other relevant factors.
Each proof of concept will incorporate key features, including but not limited to:

- Multiple Brokers in the Distributed Network, minimum 3 nodes in the cluster
- Real time multi-client(publisher/subscribers) connectivity
    - Can be physical or virtual simulated devices sending data
- Inter-broker communication and synchronization for topics and messages
    - Should handle multiple topics, shared or used by multiple devices.
- Implemented load balancing techniques that distributes messages across the different brokers.
- Implemented broker redundancy and failover measures on the developed solutions.
- Implemented security and authentication controls: encryption, authorization and data integrity checks.
- Easy deployment and setup procedure, possible through containerization.
- Source code and documentation will be made available.

Performance monitoring tools

To effectively evaluate and compare the performance of different distributed MQTT broker architectures, **performance monitoring tools** will be developed and integrated into the system. These tools will enable **real-time tracking, analysis, and visualization** of key performance metrics. The following will be developed to help monitor and evaluate system performance:

- **Real Time performance dashboard**
    - A command line interface or web-based tool for monitoring the MQTT broker cluster.
    - Displays different metrics need to evaluate performance like latency, message throughput, resource utilization etc.

- Possible graphical representation for load balancing in real time.
- **System logging tool**
  - Logs all actions being performed in the cluster. This will be helpful in the case of failure.
  - Also logs client connections and activity.
  - Alerts/alarms are triggered if an error occurs, and a message is sent to the administrator.

Hardware

The developed solution will most likely be **containerized and deployed in virtual environments** meaning **no physical hardware deliverables** will be expected for the setup and implementation of the distributed MQTT cluster network. If deployment requires **multiple host machines**, detailed **setup instructions** will be provided for configuring **local or cloud-based broker-to-broker communication**. While the research does not mandate physical IoT devices, the organization will provide access to **physical client devices** if necessary for **testing system functionality**. These devices will only be used during the research phase and will be made available for **final demonstrations** as needed.

Documentation

Project Plan

This will be the document outlining the project goals, objectives, activities and plan of implementation. This document is the initial plan at the start of the internship and can be changed or improved on as the project progresses. The project plan will be drawn up and submitted by the third week of the internship.

Technical Report

This will be a detailed research report covering the initial research, component research, design phase, implementation steps and testing results. It will detail all the work that is being done during the internship assignment. The document will also detail any short comings and recommendations that might help improve the final product in the future. This document will be submitted in 3 versions:

- **Draft Version:** includes the structure of the report and the complete chapter 1
- **Concept version:** includes the draft report and concept version of the content of all the other chapters >80% done.
- **Final version:** This version is 100% complete

Learning report

This document contains a reflection on the methodology used, a self-reflection, a self-analysis and a personal development plan. This document will also be submitted in 3 versions:

- **Draft Version:** includes the structure of the report and the complete chapter 1
- **Concept version:** includes the draft report and concept version of the content of all the other chapters >80% done.
- **Final version:** This version is 100% complete

Presentation

The researcher will be expected to give periodic presentation internally to the stakeholders within the organisation and/or supervisor, to update them on the progress or give a demonstration of the developed solution. Additionally, there will be a final demo and presentation at the end of the internship where the researcher will show the final product that has been developed to the stakeholder and all others interested in attending. There will also be a final presentation at the end given at the researcher's university which will serve as an assessment for the project objectives and work archived.

Quality Control

This section outlines the measures taken to maintain **code quality, correct implementation, testing, and validation of research results**. Ensuring the quality of both **intermediate and final project outcomes** is crucial for the success of this research assignment and will be accomplished through a vigorous review process, iterative development, testing protocols, and validation procedures

Research Goals: SMART Method

To ensure that project objectives are well-defined and measurable, the **SMART (Specific, Measurable, Achievable, Relevant, Time-bound) method** will be used.

- **Specific:** The goal should be clear and specific and answer what the researcher wants to accomplish [13].
- **Measurable:** What data can be used to evaluate the outcomes of the goal?
- **Achievable**: With the resources and time available, can the goal be accomplished?
- **Relevant**: Does the goal align to the overall objective of the project or research? How will it influence the result?
- **Time Bound:** Set a deadline for when the goal will be accomplished.

Each goal mention in Section: "Problem Analysis and Project Objectives" will be refined further in the table below:

Table 3: SMART Analysis

Goal	Specific	Measurable	Achievable	Relevant	Time Bound
How is the performance of a centralized MQTT broker system, and what are its limitations?	Develop and analyse the latency, throughput, and failure points of a traditional centralized MQTT system.	Compare performance using benchmarking tools.	Can be developed or with the use of existing MQTT broker used by the organization.	Provides a baseline comparison that can be used to justify success of project solution.	One of the first implementation task of the project. With an SLE of 7 days .
What are the fundamental designs currently developed that achieve an effective distributed MQTT broker system, and how do they differ in terms of message routing, load balancing, and fault tolerance?	Identify at least three distributed MQTT architectures	Conduct a comparative analysis of message routing, load balancing, security and fault tolerance mechanisms.	Use literature review and existing open-source implementations for reference.	Essential for determining the most effective architecture to develop.	Conducted during the initial research tasksbefore implementation starts.
What scalability strategies can be integrated into a distributed MQTT architecture to efficiently handle high traffic loads and a growing number of connected devices?	Investigate horizontal and vertical scaling techniques in distributed broker systems.	Evaluate latency, message queue size, and broker load balancing efficiency.	Implement load balancing and broker scaling mechanisms in the proof of concept.	Ensures the system can handle large-scale IoT deployments.	Completed during the implementation and testing phases.
How can session persistence and state synchronization be managed across multiple brokers in a distributed system?	Explore broker replication, message state tracking, and failover synchronization techniques.	Measure the effectiveness of synchronization mechanisms in maintaining data consistency.	Implement the mechanisms in the proof of concept.	Ensures reliability by preventing message loss or duplication.	Completed alongside fault tolerance testing in the testing phase.
What performance benchmarks and metrics should be used to evaluate the effectiveness of a distributed MQTT broker architecture?	Define key performance indicators (KPIs) including latency, throughput, fault tolerance, and resource utilization.	Develop a testing framework using performance monitoring tools.	Use tools like Grafana and MQTT stress testing tools to measure results.	Ensures the project follows industry standards for performance validation.	Benchmarking should be finalized before completing the final performance evaluation .
What security considerations must be accounted for when deploying a distributed MQTT broker system?	Identify authentication, encryption, and network security challenges in distributed MQTT deployments.	Implement and evaluate security features for encryption, authentication, and access control.	Use penetration testing tools and security logging to validate system protection.	Security is critical to prevent unauthorized access and ensuring data integrity.	Considered during each coding implementation and integrated before testing.

Implementation

Code and system designs will be reviewed periodically with feedback from the **supervisor and available research group members**. The goal of the reviews will be to ensure **code correctness, maintainability and adherence to professional coding practices**. **Version control** of the coding solutions will also be implemented with the use of **Git** to help with quality control, and potential roll backs if errors occur. The project has also been split into smaller and more manageable tasks that will be executed and tracked through a **kanban workflow, which encourages continuous improvement based on feedback, test results and refinement.** Task will continuously be reviewed to ensure that they are still relevant to the project completion or if further refinement is required. Additionally, stakeholder meeting and demos will be held periodically where feedback on the solution and recommendations will be welcomed which will help with production of a stakeholder approved product.

The documentation produced during the project will be subject to multiple rounds of review, ensuring quality and continuous improvement. Each document, including README files will be **peer-reviewed and continuously refined based on feedback** from the **internship supervisor, university coach and peers** to improve clarity. These iterative changes will also reflect changes made through the project.

Testing and Validation Procedure

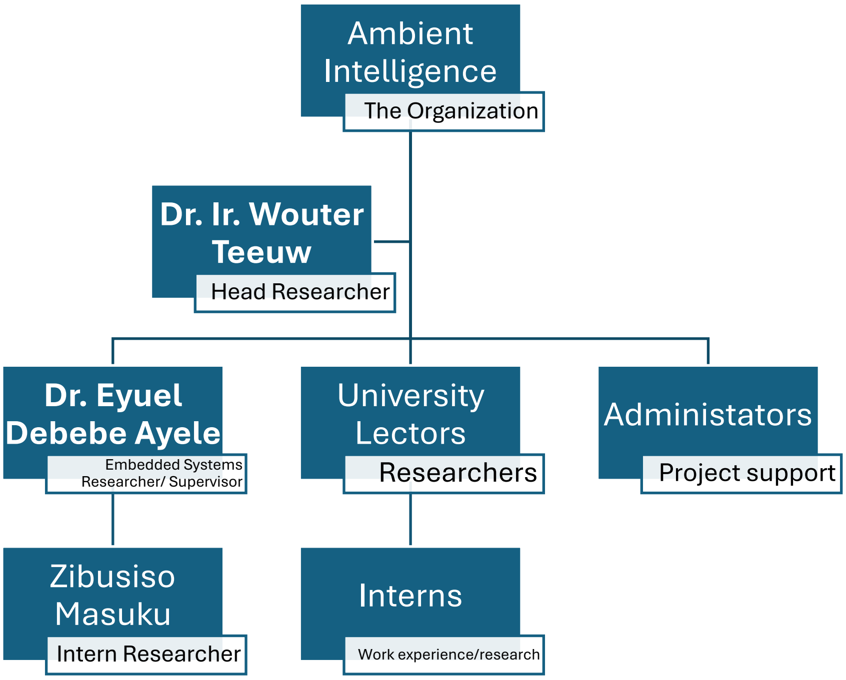
Quality control during testing will be maintained through **clearly defined test criteria, automated testing mechanisms, and real-time performance monitoring tools**. The test criteria will be defined under 3 categories:

- **Functionality**: verifying that all functions like message routing, synchronisation and failover mechanisms are working as expected.
- **Performance:** Measuring the KPIs set to ensure that the system is functioning as expected.
- **Security Testing:** System security can be tested by performing simulated cyber-attacks, which the researcher has some experience on, and will help ensure the integrity of the system.

Project Organization

This section outlines the roles and responsibilities in the project. It describes how the student, supervisor, and research team collaborate to complete the study.

Research Group Structure



The researcher will conduct this assignment as an intern within the organization, under the direct supervision of **Dr. Eyuel Ayele**, now referred to as **the supervisor**. The supervisor will provide guidance in **planning and executing the project**, ensuring alignment with research objectives. Meetings between the **researcher and supervisor** will be held **bi-weekly** to review progress and discuss next steps, following an **Agile approach**.

Stakeholders

Other **researchers within the organization** will be considered **project stakeholders**, offering **feedback and assistance**as needed throughout the research process. Additionally, the researcher will collaborate with **other interns on a day-to-day basis**, providing mutual support when required. In addition, the researcher will participate in **weekly organizational meetings**, where all researchers engage in **knowledge-sharing sessions** to exchange insights and feedback on ongoing projects.

Researcher's Responsibility

The researcher's primary focus will be on the **"Distributed MQTT Broker Architecture"** project and will not be expected to contribute to **other projects or influence the work of other researchers or interns**. For administrative or cost-related matters arising during the project, designated **contact persons** within the organization will be available to provide the necessary support.

Working contract

The researcher is required to log work hours on a **weekly basis** and is expected to maintain a **40-hour workweek**. The work arrangement follows a **hybrid model**, with a minimum requirement of **three days per week** in the office to facilitate collaboration and supervision.

Planning

This section presents the project timeline, and outlines the key phases, tasks and how the methodology used ensures that the deadlines are met.

Project Timeline and Planning

The research assignment spans **20 weeks (February 4th – June 22nd, 2025)** and is structured into **phases that align with research goals and agile methodology working style**. The **Gantt Chart** (Appendix 1) outlines the **progression of activities, milestones, and deadlines**. The planning accounts for **decision-making delays**, dependencies on feedback from the supervisor and stakeholders, and adjustments based on testing results. An SLE of 7 days has been set based on all the different variables mentioned for how long tasks should take. The tasks are tracked through **a Kanban board**, where items progress from **"Backlog" to "In Progress," "Review," "Testing," and finally "Done."** The flexibility of this approach ensures that **high-priority tasks are completed first**, while unfinished or blocked tasks are continuously reassessed.

Key Deadlines

All key deadlines of the project have been represented in the table below:

Table 4: Key Deadlines

Deliverable/Milestone	Deadline
Project Plan	Week 3
Literature Review & Initial Research	Week 3
First Company Visit	Week 5
Implementation of Basic Centralised MQTT Broker for Benchmarking Purposes	Week 6
Draft reports (Technical, Learning Reports)	Week 9
Implementation of key feature in the Distributed Architecture Systems	Week 11
Concept Reports (Technical and Learning Report)	Week 13
Bench Marking Tools/Scripts completion	Week 13
Company Visit Go-no-go	Week 15
Final Reports (Technical and Learning Reports)	Week 17
Implementation of Supplementary features on the chosen solution	Week 19
Company Assessment to Saxion	Week 19
Final Presentations	Week 20

Cost and Benefits

This section evaluates the required resources, project costs, and expected benefits. It weighs the investment against the improvements in IoT scalability and reliability.

Costs

The main factors of the project will involve the following:

Hardware Costs

At the time of drafting this project plan, **no additional physical hardware expenses** are anticipated. Most required hardware, such as **sensors and IoT devices**, is available in the organization's lab. The researcher will use a **personal computer** for development, minimizing hardware-related costs.

Should the implementation require **additional physical hardware**, the researcher will consult with the **supervisor to assess feasibility and request funding** if deemed necessary. Additionally, if a **cloud-based deployment** is incorporated into the project, **potential costs may arise**, depending on the selected platform and resource usage.

Software and Network infrastructure Costs

The project will primarily utilize **open-source software and MQTT broker tools** for development, testing, and deployment, eliminating the need for **licensing costs**. Network infrastructure costs will be **minimal**, as most testing will be conducted in a **virtualized or local environment**. However, if **cloud-based deployment** is explored, potential network usage fees may be incurred.

Time and Internship Compensation

This assignment will be executed over **20 weeks**, with the researcher working **40 hours a week**. As part of the internship agreement, the researcher will receive a **monthly compensation of €475**, allowing the researcher to dedicate the required time and effort to the successful execution of the project.

Benefits

The research will contribute to **scalable, resilient, and secure IoT communication** by addressing the limitations of **centralized MQTT brokers**. The developed **distributed architecture** will aim to improve **performance, fault tolerance, and load balancing**, ensuring efficient communication in **high-traffic environments**. Additionally, the project will provide **practical implementations, benchmarking reports, and security enhancements**, offering valuable insights for **future research and real-world applications**. Ideally the final proposed solution should be immediately scalable and applicable to the organisation's needs.

Risk Analysis

This section identifies potential risks in the project. It covers technical risks, time management issues, and personal delays like illness. The risks will be weighed and mitigated if possible and accounted for during planning.

Risk Analysis Model Results

A risk analysis model created by Jurgen Winkel was used to calculate and evaluate the potential risks affecting the project's success [14]. It assesses key factors like time management, complexity, project planning and project management. It also accounts for the researcher and team's ability and knowledge into the subject matter. By assigning scores to each category, the model calculates a risk percentage, helping to identify areas that require mitigation. If the project's Risk Percentage exceed 50%, the project should not proceed and needs additional refinement.

		Total		**76**
		Risk percentage ***		**17,55%**

** Value chosen by Zibusiso Masuku(Researcher)*
*** Defined factor in this model.*
**** Risk percentage is total score divided by 433 (maximal score) times 100.*

The risk percentage provides an overview.
If the percentage is higher than 50%, the project should not carried out in this form.
The next table shows the risk percentage per catagory.

Category (maximum score versus calculated score)				
Time factor	Maximal	40	Score	24
Complexity	Maximal	80	Score	18
Researcher Ability	Maximal	65	Score	3
Project management	Maximal	129	Score	21
Project definition	Maximal	119	Score	10

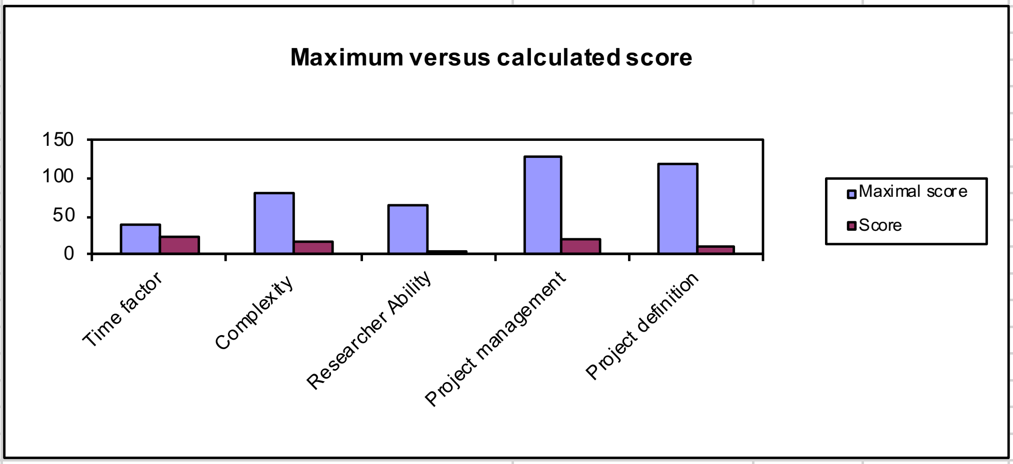


Figure 5: Risk Analysis Extract

The researcher filled in the values for the model and a risk analysis score was calculated and a graphical data output given. An extract showing the result is shown in figure 5 and the full risk analysis can be found under Appendix 2: Risk Analysis Model. The total risk percentage for this project is **17.55%**, indicating a **low-risk level** and suggesting that the project is planned well with minimal risk to its success. The **time**

factor is the most significant risk; however, it has been mitigated with good project planning, introduction of SLE's and prioritization of tasks with the methodology being used to help ensure that all important tasks are implemented within the project timeline.

While **the project maybe be complex**, the **refinement** of the task to more manageable tasks has helped to **significantly reduce the risk** of the project not meeting the goal. Additionally, the iterative approach ensures that there is a finished product at the end even if some less important tasks aren't completed. The researcher's ability at the start has a low-risk score, reinforcing confidence in the project's feasibility.

Furthermore, there are certain risk that have not been accounted for by the model and cannot be mitigated like illness. These risks have low chances of occurrence and will not significantly affect the project progress. The risks are there for accepted and will be handled as they arise.

Sources

[1] Ambient Intelligence , “Ambient Intelligence Research Group,” Saxion University of Applied Sciences, 2025. [Online]. Available: https://www.saxion.nl/onderzoek/smart-industry/ambient-intelligence. [Accessed 6 February 2025].

[2] A. T. H. W. S Ding, “Opportunities and risks of internet of things (IoT) technologies for circular business models: A literature review,” *J. Environ. Manag,* vol. 336, no. 117662, 2023.

[3] A. M. R Mishra, “Current research on Internet of Things (IoT) security protocols: A survey,” *Computers & Security,* vol. 151, no. 104310, 2025.

[4] P. P. R. H. N Saha, “Performance Evaluation Framework of MQTT Client Libraries for IoT Applications in Manufacturing,” *Manufacturing Letters,* vol. 41, p. 1237–1245, 2024.

[5] J. H. K. H. A Dohyun, “Enabling low-latency digital twins for large-scale UAV networks using MQTT-based communication framework,” ICT Express, 14 January 2024. [Online]. Available: https://doi.org/10.1016/j.icte.2025.01.001. [Accessed 6 February 2025].

[6] A. R. E Longo, “Design and implementation of an advanced MQTT broker for distributed pub/sub scenarios,” *Computer Networks,* vol. 224, no. 109601, 2023.

[7] A. M. e. al., “Agri-tech innovations for sustainability: A fire detection system based on MQTT broker and IoT to improve environmental risk management,” *Results in Engineering,* vol. 24, no. 103683, 2024.

[8] R. D. e. al., “Distributed MQTT Broker: A Load-Balanced Redis-Based Architecture,” *International Conference on Emerging Smart Computing and Informatics (ESCI),* vol. 10, no. 1109, pp. 1-6, 2024.

[9] M. Davidson, “Unpacking MQTT’s Limitations: A Centralized Data Storage Dilemma,” Medium.com, 27 November 2023. [Online]. Available: https://medium.com/@markjoulex/unpacking-mqtts-limitations-a-centralized-data-storage-dilemma-483c5756932b. [Accessed 7 February 2025].

[10] M. A. Spohn, “On MQTT Scalability in the Internet of Things: Issues, Solutions, and Future Directions,” *Journal of Electronics and Electrical Engineering,* vol. 10, no. 37256, 2022.

[11] “Exploring the Basics of MQTT Broker Clustering: An Introduction,” EMQ Technologies Inc, 2 December 2024. [Online]. Available: https://www.iotforall.com/exploring-the-basics-of-mqtt-broker-clustering-an-introduction#. [Accessed 7 February 2025].

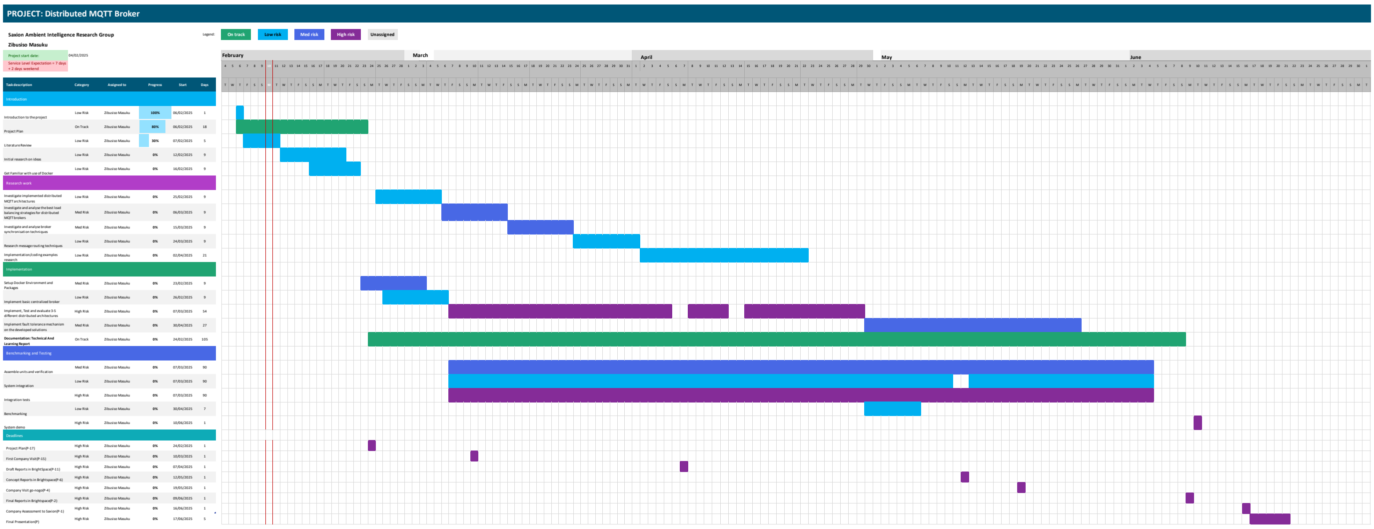
[12] Z. L. e. al., “A security-enhanced scheme for MQTT protocol based on domestic cryptographic algorithm,” *Computer Communications,* vol. 221, pp. 1-9, 2024.

[13] Mind Tools, “SMART Goals,” Mind Tools, 2025. [Online]. Available: https://www.mindtools.com/a4wo118/smart-goals. [Accessed 11 February 2025].

[14] R. Grit, Project Management: A Practical Approach, Groningen: Noordhoff Uitgevers bv, 2011.

Appendices

Appendix 1: Gantt Chart



Appendix 2: Risk Analysis

	Risk Analysis	Print date			
	MQTT Broker Architecture Research	*11/02/2025*			

	Risk	Value *	Factor **	Weight ***	Total risk
Time factor		↓choose one↓			
1	Estimated duration of project	6+ months	3	4	12
2	Does the project have a definite deadline?	Yes	2	4	8
3	Is there enough time to complete the project within the time permitted?	enough	1	4	4
Complexity of project		↓choose one↓			
4	Number of functional subsectors involved	1	0	4	0
5	Number of functional subsectors that will make use of the results of the project	1	0	2	0
6	Is is a new project or one that will be adapted?	New project	3	5	15
7	To what extend do current authorizations in the organization have to be adjusted?	Not	0	5	0
8	Are other projects dependent on this project?	No	0	5	0
9	How are users (of the project results) likely to respond to it?	Enthousiastic	0	5	0
10	Is the project broken down into phases and does the project depend on coordination between them?	No	1	3	3
The project group					
11	Where do the project workers come from?	Mainly internally	0	4	0
12	Where is the project located?	1 location	0	2	0
13	How many project members work for more than 80% at peak hours?	1-5	0	5	0
14	Balance between subject experts and project experts	Good	0	5	0
15	Are users involved in the project?	to a reasonable extend	1	3	3
Project management		↓choose one↓			
16	Does the project management team have any knowledge of the subject?	a lot	0	3	0
17	Does the project management have any knowledge of how to plan a project?	a lot	0	3	0
18	How much experience does the project manager have with a project like this?	a reasonable amount	1	3	3
19	Does the adviser have much knowledge of the field of the project?	a lot	0	5	0
20	Do the subject experts have much knowledge of the field?	a lot	0	5	0
21	How involved are responsible managers in the project?	reasonably involved	2	5	10
22	Is there any chance that the project team will change during the project?	little chance	0	5	0
23	Is the project team using existing methods or are they creating their own tools?	some existing methods	2	4	8

Risk analysis continued ….

	Risk	Value *	Factor **	Weight **	Total risk
Project definition		↓maak keuze↓			
24	Are project members sufficiently aware of problems and objectives?	yes, everybody	0	5	0
25	Is the field of result (scope) sufficiently defined?	yes	0	5	0
26	Is there enough distinction between this project and other projects?	considerable	0	4	0
27	Has enough time been resereved for coordination and decision-making?	considerable	0	4	0
28	Are the boundaries and preconditions clear?	yes	0	4	0
29	Are the boundaries limiting enough?	moderately	2	5	10
			Total		**76**
			Risk percentage ***		**17,55%**

** Value chosen by Zibusiso Masuku(Researcher)*
*** Defined factor in this model.*
**** Risk percentage is total score divided by 433 (maximal score) times 100.*

The risk percentage provides an overview.
If the percentage is higher than 50%, the project should not carried out in this form.
The next table shows the risk percentage per catagory.

Category (maximum score versus calculated score)				
Time factor	Maximal	40	Score	24
Complexity	Maximal	80	Score	18
Researcher Ability	Maximal	65	Score	3
Project management	Maximal	129	Score	21
Project definition	Maximal	119	Score	10