35th Annual INCOSE international symposium
hybrid event
Ottawa, Canada
July 26 - 31, 2025

Redesigning Systems Architecture for AWS Platform Migration: A Case Study of an Energy Monitoring System

Catalina Klarissa Mae Tagavilla-Gaza
University of South-Eastern Norway
klarissagaza@gmail.com

Henri Giudici
University of South-Eastern Norway
henri.giudici@usn.no

YangYang Zhao
University of Oslo
yangyang.zhao@sfe.uio.no

Copyright © 2025 by Catalina Klarissa Mae Tagavilla-Gaza, Henri Giudici, YangYang Zhao. Permission granted to INCOSE to publish and use.

Abstract. Monitoring activities play an important role in increasing the effectiveness of the energy adopted in industry. Accurate tracking in combination with analytical usage patterns helps to identify potential inefficiency in the adopted setting. Cloud-based services are adopted to help with these monitoring activities. Energy companies adopt these platforms to leverage real-time energy data. Recent advancements in this field have driven companies to continuously evolve and develop their energy platforms. Platform migration is important for improving scalability, flexibility, and advanced analytics. This study presents an industrial case on redesigning system architecture for platform migration. The case company aims to modify its current Energy Monitoring System (EMS) by shifting from its current EMS to a novel platform. The study showcases both technical and organizational challenges involved in integrating the novel platform into the existing system architecture. It then proposes a strategy to minimize stakeholders' resistance and ensure an easy platform migration aligned with the company's objectives.

Keywords. Energy mapping, Energy monitoring, Platform migration, System architecture, System engineering.

Introduction

Energy consumption has witnessed exponential growth in the recent era of rapid industrialization and urbanization. Energy monitoring is essential for businesses as it provides real-time data on energy consumption. Such monitoring enables them to optimize energy, increasing their efficiency and sustainability. Energy consumption has a strong positive correlation with CO_2 emissions. Over 40% of the CO_2 emissions are due to the burning of fossil fuels for energy consumption. Managing the carbon footprint, which includes all direct and indirect greenhouse gases from business activities, is crucial for reducing environmental impact and aligning with global sustainability efforts (Muller and Giudici, 2024). Energy consumption is a significant driver of carbon emissions and a major contributor to climate change. Improving energy efficiency helps companies reduce their environmental impact and energy costs, providing financial benefits and enhancing value for stakeholders (Gaza et al., 2024).

Platform migration plays a crucial role in achieving these goals, as companies aim to improve scalability, flexibility, and advanced analytics. Information Technologies (IT) play a significant role in improving EMS through real-time data collection, advanced analytics, and effective communication with stakeholders. This study focuses on a situation where the Company is migrating from one cloud service provider (StackHero) to Amazon Web Services (AWS). Platform migrations are inherently complex tasks and comprise multiple activities performed at several technical and organizational levels. There are risks associated with each of these tasks. Each risk has significant potential to impact several parameters, such as technical performance, data security, operational continuity, stakeholder engagement, and financial outcomes. We take risk identification and management into consideration in order to ensure a smooth transition, aligning platform migration with the company's long-term strategic objectives. By adopting system engineering methodology, we fulfill the main research purpose of redesigning system architecture for platform migration.

The Case Company and Project

We performed the case study in collaboration with Semcon, henceforth referred to as the Company. The Company's expertise is in energy metering, engineering, digital services, and sustainability solutions (Semcon, 2024). The development of an EMS is one of the Company's projects. This project showcases the Company's commitment to integrating advanced technology with user-focused design, as illustrated in Figure 1. The EMS project gathers real-time energy data from multiple sources to optimize energy use and management. The primary devices employed are the e-gauge meter and Homey Pro. The e-gauge meter monitors electrical and solar consumption, allowing users to track and enhance their energy consumption effectively. Meanwhile, Homey Pro acts as a smart home controller, integrating home devices to enhance energy management. A router transmits all data collected from these devices to StackHero's cloud computing service. This platform handles real-time processing of large-scale data, analyzing and transforming raw data into actionable insights. These insights are then displayed on a user interface dashboard, providing clear and helpful information for users to decide about their energy use. The planned migration will enable the Company to leverage a broader suite of cloud-based services and infrastructure, enhancing overall performance.

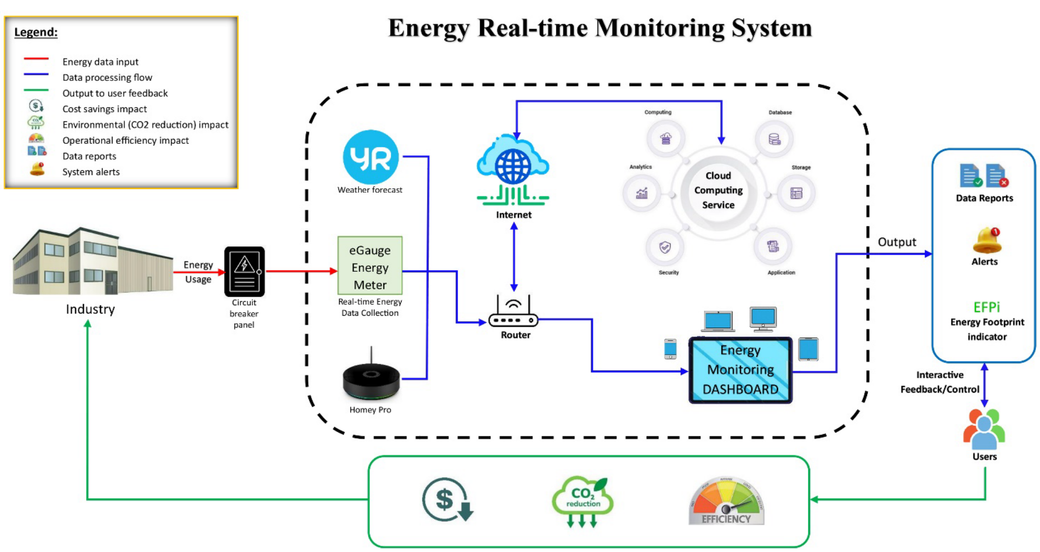

Figure 1. Schematic Overview of the EMS

Challenges of Platform Migration

The migration from StackHero to AWS involves replacing current system components with those from AWS. This change calls for thorough testing and quality assurance to guarantee smooth operation in the new environment. Additionally, key project stakeholders must tackle both technical and organizational challenges to ensure the migration is successful (Kolb et al., 2015; Khajeh-Hosseini et al., 2010; Mukherjee, 2019).

The migration to AWS presents several technical challenges. System architecture needs to be adapted to integrate seamlessly with AWS's services (Dabbagh et al., 2018). As such, managing virtual machine (VM) live migration is crucial, as it involves transferring VMs from one physical server to another within cloud data centers without causing downtime or disrupting services (Reguri et al., 2016). It is also important to maintain database integrity during the transition (Engelsrud, 2019). Moreover, scalability and performance optimization are necessary, impacting customer service-level agreements (SLAs) (Samrajesh and Gopalan, 2013). Ensuring compliance with security standards and regulatory frameworks adds another layer of complexity to the migration process (Son et al., 2017).

From an organizational perspective, change management is a big concern. Such change necessitates implementing structured communication and training programs to ease the migration of these employees (Fowley et al., 2017; Engelsrud, 2019). Another significant challenge is addressing skill gaps, which may require employee training (Zheng and Du, 2014; Shastry et al., 2022). Additionally, organizations need to manage resources to ensure budget, personnel, and infrastructure align with the goals of migration (Agavanakis et al., 2012; Son et al., 2017). Adhering to compliance and governance is also necessary, requiring policies and monitoring mechanisms to comply with data protection laws and regulations (Mwansa and Mnkandla, 2014; Yang and Wan, 2020). Lastly, effective collaboration and communication across departments and with stakeholders are essential to prevent operational silos during the migration (Reguri et al., 2016; Yang et al., 2019).

Research Question

Establishing a sound system architecture for the new cloud platform is essential, as it forms the foundation for the migration process. This study thus aims to address the challenges of migrating the platform to AWS by redesigning a compatible system architecture. The new system architecture is to meet the foundational requirements compatible with the target platform (i.e. AWS), while also managing and mitigating the risks associated with this migration. There are several notable risks, including technical compatibility issues, data security vulnerabilities, operational downtime, stakeholder resistance, and cost overruns. Addressing these risks early in the process will support a smooth and successful migration by helping to meet stakeholder needs while maintaining system integrity. Therefore, our research question guiding this study is: *"How to redesign the system architecture to ensure compatibility with the target migration platform (AWS) and stakeholder satisfaction?"*

Literature Review

This section introduces the state-of-the-art of platform mitigation and the knowledge applied to solving the research question. The review helps readers understand the challenges, identify the process of platform migration, and identify existing solutions.

Platform Mitigation

Platform migration involves transferring any digital information and activity to cloud environments, often requiring code adjustments and variations in deployment (Fowley et al., 2017). As the Company undertakes the EMS migration from StackHero to AWS, the process can be complicated and requires structured planning and precise execution. Kolb et al. (2015) emphasized the challenges of migrating across multiple cloud platforms, while Engelsrud (2019) discussed the *Lift and Shift* methodology for transitioning applications to Oracle Cloud. Zheng and Du (2014) suggested a structured approach for migrating client-server applications to AWS.

EMS. EMS is essential for tracking energy consumption and requires accurate data collection along with real-time monitoring. Redesigning the EMS architecture to leverage platform migration not only enhances functionality and performance by providing scalable infrastructure and improved data security but also boosts advanced analytics capabilities (Dabbagh et al., 2018). Such migration improves connectivity with Internet of Things (IoT) devices, fostering cross-system optimization and enabling resource scaling based on demand, which enhances scalability and ensures reliability through built-in redundancy and failover mechanisms (Pahl and Xiong, 2013). Moreover, geographic redundancy from globally distributed data centers bolsters reliability, guaranteeing uninterrupted operation even during regional outages (Dabbagh et al., 2018).

AWS for EMS. Integrating AWS services like IoT Core, Timestream, and QuickSight into the EMS improves operations and introduces innovative solutions. AWS includes many services designed to function cohesively and enhance cloud applications. *AWS IoT Core* is a managed cloud service that lets connected devices easily and securely interact with cloud applications and other devices. *AWS Timestream* is a time-series database service that efficiently handles high throughput demands, enabling fast storage and retrieval of time-series data at scale for EMS. *Amazon QuickSight*, a business intelligence service, utilizes machine learning to provide advanced analytics for building visualizations and performing ad-hoc analysis. These components are not independent but are part of the comprehensive AWS ecosystem designed to provide a seamless and integrated experience that supports the specific needs of EMS, making it a preferable choice over competitors like Azure and Google Cloud (Khajeh-Hosseini et al., 2010; Mukherjee, 2019; Olariu, 2023).

Challenges and Impacts

Technical Challenges of Migration. Technical challenges in platform migration include application and infrastructure design, security and compliance, reliability, scalability, and the need for continuous upgrades and testing. Yang et al. (2019) discussed the challenges of integrating existing systems with cloud architectures to maintain operational continuity. Zheng and Du (2014) pointed out that security is a major concern, emphasizing the need for robust data protection and adaptive security measures. Reliability and fault tolerance are critical, as Samrajesh and Gopalan (2013) noted, and supported by frameworks suggested by Fowley et al. (2017). Engelsrud (2019) also addressed scalability challenges, stressing the importance of performance maintenance during scaling processes. Gannon et al. (2017) further emphasize the importance of seamless upgrades and compliance adherence throughout the migration.

Organizational Challenges and Strategic Impacts. Addressing organizational challenges during platform migration involves strategic planning, training, resource allocation, and improving collaboration. Strategic planning is crucial to align migration efforts with organizational goals and involve all stakeholders, which helps manage risks and achieve strategic objectives (Orue-Echevarria et al., 2011; Yang and Wan, 2020; Ranganathan and Sampathrajan, 2023). Training is necessary for bridging skill gaps and ensuring a smooth transition (Samrajesh and Gopalan, 2013; Yang and Wan, 2020) and continuous learning by (Fowley et al., 2017). Moreover, enhancing collaboration and communication also helps migration success by facilitating smoother transitions and reducing resistance to new technologies (Agavanakis et al., 2012; Samrajesh and Gopalan, 2013; Yang et al., 2019). It is notable that resource allocation is essential in supporting migration efforts (Samrajesh and Gopalan, 2013; Abernathy et al., 2020; Yang and Wan, 2020).

In particular, organizations need to consider the financial impacts, focusing on potential cost savings and the return on investment (ROI), which are crucial for justifying migration (Olariu, 2023).

Environmental Impact and Sustainability. Migrating the EMS to a platform that utilizes energy-efficient data centers and engages in renewable energy initiatives could help reduce the carbon footprint and enhance environmental responsibility (Mukherjee, 2019). The migrated platform can collect and analyze real-time energy usage data by integrating sensors, data acquisition systems, and analytics software. This capability allows for identifying patterns, trends, and anomalies in energy consumption, facilitating more informed decision-making and promoting sustainable energy use (Yang et al., 2019).

Systems Engineering and Architecture

Systems engineering plays a critical role in complex project management by providing a structured approach to designing, integrating, and optimizing systems – an essential foundation for the successful platform migration (SEBoK, 2023). It helps anticipate and address potential challenges and supports creating scalable and efficient platforms. Complementing this, systems architecture is a strategic asset, ensuring effective energy monitoring and adaptability to future demands through platform migration. According to the ISO/IEC 42010-2011 standard, architecture involves the "fundamental concepts or properties of a system in its environment embodied in its elements, relationships, and in the principles of its design and evolution" (Pineda, 2023, p. 595). Utilizing Architectural Block Diagrams (ABDs) and Functional Block Diagrams (FBDs), which depict a system's high-level structure and how its components interact, respectively, helps in managing the complexity of the existing systems and guiding the architectural changes necessary for migration (Hilburn, 2023, p. 956–959).

Many studies have underscored the significance of integrating systems engineering with platform migration strategies to develop a functional and adaptable systems architecture. However, empirical studies on their practical application remain scarce, particularly in the context of migrating EMS. Several case studies from industry leaders like Netflix, Airbnb, Dropbox, and Capital One demonstrate the value of such integration, offering insights into architectural redesigns for cloud adoption and innovative solutions for data security and regulatory compliance (Khajeh-Hosseini et al., 2010; Kolb et al., 2015).

Data and Methods

Research Design

This section outlines the research design for this single case study. The research design employs a systems engineering methodology to solve the research question, as shown in Figure 2. It includes four main phases. Initiation Phase sets the project in motion with a comprehensive problem statement that clearly shows the challenges and objectives of migrating the EMS. We identify key stakeholders to ensure that the project considers the needs and expectations of all parties. This step is essential for aligning the project's objectives with the stakeholders' requirements. The analysis phase involves an extensive review of the current system architecture to prepare for redesign. The review includes a detailed literature study focusing on AWS-specific migration challenges, accompanied by preliminary interviews and on-site visits. These activities provide an understanding of the AS-IS system architecture and operational environment. To ensure a successful migration, the research incorporates a risk management framework at every stage of the process. The framework includes identifying potential risks, assessing their impact and likelihood of occurrence, and developing mitigation strategies. The System Redesign Phase is an iterative process that involves developing the proposed TO-BE system architecture, incorporating the insights gathered from the analysis. This phase includes a rigorous redesign analysis to determine the best features for inclusion in the new system. The research activity involves continuously seeking stakeholders' feedback to refine the design, ensuring the new system architecture meets the evolving needs and anticipates future scalability requirements. The Validation Phase ensures that the TO-BE system architecture adheres to all specified requirements through feedback from key project stakeholders. The case study project concludes with a thorough review of the outcome against the original goals, making necessary adjustments based on feedback and documenting the results to confirm the strategic alignment with the broader organizational objectives. After completing the validation, the project identifies areas for future research and provides practical recommendations based on the insights gained during the platform migration.

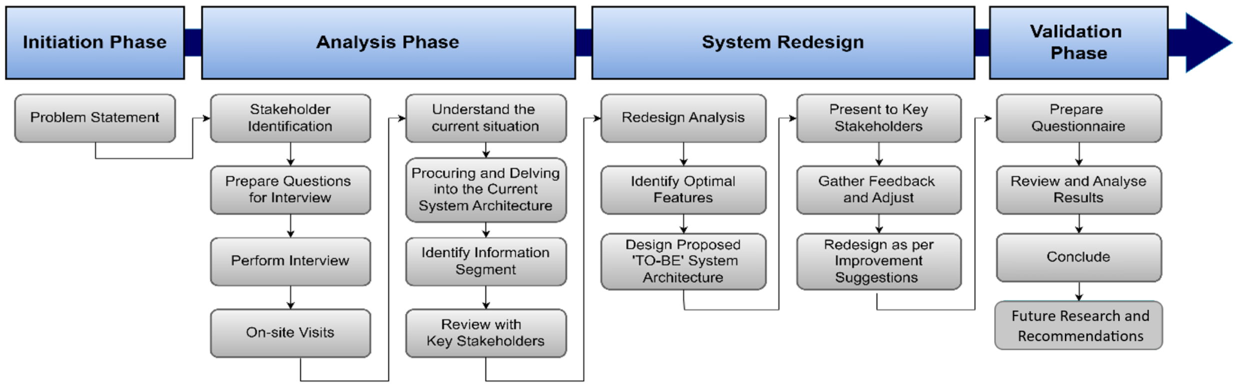

Figure 2. Research Design

Data Collection

This study used multiple methods to collect the case data, including interviews, on-site visits, a presentation workshop and a survey. Each method was chosen to address distinct aspects of the system analysis and to facilitate meaningful stakeholder engagement. This comprehensive approach enabled a multifaceted examination of the EMS, leading to reliable and insightful findings.

Interviews. Interviews were conducted in preliminary and structured formats to gather insights into the Company's migration process. *Preliminary interviews* aimed to understand the Company's EMS approach and an overview of the EMS, such as energy usage, energy reduction strategies with the e-gauge meter, components integration, etc. The data also includes the Company's EMS challenges, setting a foundation for more in-depth investigation in subsequent interviews. *Structured Interviews* geared for more detailed information on the EMS and its system architecture migration. Table 1 provides a summary of the interviews conducted, including the interview time, the purpose and the information of interviewees.

Table 1. Interview Types, Purposes and Participants

No.	Type	Date	Purpose	Participant(s)	Experience	Phase
PI 01	Preliminary Interview (Phone)	22.09.2023	Understand the Company's EMS approach	Business Development Manager	14 years, 5 months	Initiation Phase
PI 02	Preliminary Interview (Personal)	20.10.2023	Overview of the Company's EMS and information on energy usage reduction with e-gauge meter, and integration of other components.	Business Development Manager	14 years, 5 months	Initiation Phase
PI 03		13.02.2024	Understanding the Company's EMS challenges	Head of Business Design and Systems	2 years, 8 months	Initiation Phase

	Preliminary Interview (Zoom)			Engineering Expert Group Software Developer / Project Manager	 5 years, 9 months	
SI 01	Structured Interview (Personal)	19.02.2024	Detailed explanation of the EMS and major concerns about system architecture and platform migration	Head of Business Design and Systems Engineering Expert Group Software Developer / Project Manager	2 years, 8 months 5 years, 9 months	Analysis Phase
SI 02	Structured Interview (Personal)	22.02.2024	Identification and analysis of stakeholders, their roles, needs, and concerns, along with their strategies and the challenges.	Senior Software Developer	3 years, 3 months	Analysis Phase
SI 03	Structured Interview (Phone)	27.02.2024	Discussion on proposed solution (edge computing)	Software Developer / Project Manager	5 years, 9 months	Analysis Phase
SI 04	Structured Interview (Zoom)	29.02.2024	Update on proposed solution (edge computing) and its rejection	Head of Business Design and Systems Engineering Expert Group	2 years, 8 months	Analysis Phase
SI 05	Structured Interview (Personal)	05.03.2024	Discussion on the current AS-IS system architecture and necessary actions	Senior Software Developer	3 years, 3 months	Analysis Phase
SI 06	Structured Interview (Zoom)	07.03.2024	Discussion of migration to AWS, addressing the timeline, reasons, challenges associated with the shift.	Software Developer / Project Manager	5 years, 9 months	Analysis Phase

On-site Visits. The on-site visits at the Company's workshop provided a unique opportunity for direct observation of the EMS and interactions with the operational processes and project stakeholders. During these visits, we closely examined equipment such as the e-gauge meter and the HomeyPro and understood how everything was configured and integrated. The observations were instrumental in assessing the system's performance and identifying potential enhancements. Figure 3 is a visual representation of the workshop environment during these visits. Details of on-site visits are summarized in Table 2.

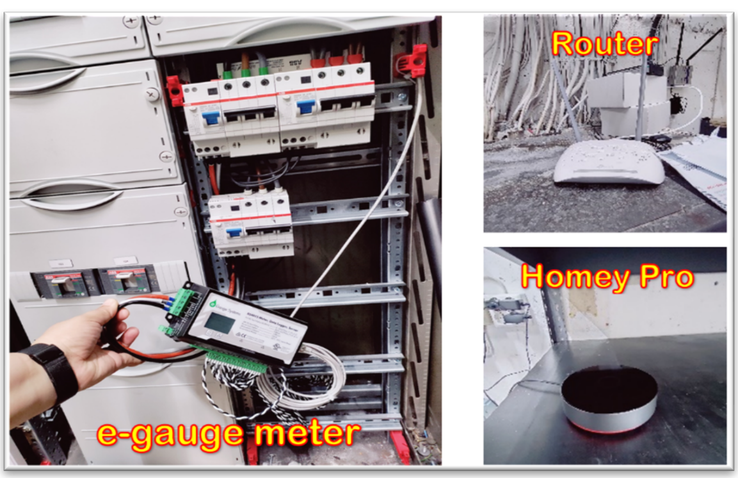

Figure 3. The Workshop Environment

Table 2. On-site Visits

No.	Type	Date/ Time	Purpose	Participant(s)	Experience	Phase
OSV 01	On-site Visit	19.02.202 4	Demonstration of the e-gauge meter integration in workshops' panel board	Software Developer / Project Manager	5 years, 9 months	Analysis Phase
OSV 02	On-site Visit	20.03.202 4	Demonstration of the integration of HomeyPro in the workshops' panel board	Senior Software Developer	3 years, 3 months	Analysis Phase

Feedback Workshop. We presented the proposed TO-BE system architecture to the key stakeholders in a collaborative presentation workshop (see Table 3). This interactive session sought detailed feedback to improve the architecture proposal. The feedback collected serves the iteration process during the system redesign phase of the final solution.

Table 3. Feedback Workshop

No.	Type	Date/ Time	Purpose	Participant(s)	Experience	Phase
Presentatio n 01	Presentation (In-Person)	20.03.202 4	Presentation of a proposed TO-BE system architecture	Senior Software Developer	3 years, 3 months	System Redesign
				Head of Business Design and Systems Engineering Expert Group	2 years, 8 months	

Survey. To complement the insights gained from interviews and on-site visits, a survey was conducted for stakeholder evaluation. The survey consisted of 17 questions, each using a five-point Likert scale to assess various aspects of the proposed TO-BE systems architecture. The objective was to quantitatively assess stakeholder satisfaction with the final solution and validate its technical soundness and operational suitability.

Case Study

This case study investigates the migration of the Company's EMS from StackHero to AWS, highlighting the strategic partnership with AWS and its use as a training platform for engineers. The migration depends on available engineering resources and is planned to ensure system continuity and minimize risks by building a parallel system. Despite potential challenges, the study analyzes the migration's effectiveness to predict energy usage while ensuring compatibility with AS-IS system architectures and stakeholder needs. Risks associated with the mentioned migration are analysed in detail. Risk identification was done through stakeholder interviews, surveys, and on-site observations, and subsequent suitable mitigation strategies were developed.

Key Stakeholders and Needs

The data from the structured interviews, particularly the one with the Company's Senior Software Developer (i.e., SI 02, Table 1), helps map the stakeholders and identify their needs in the migration process.

Stakeholder Mapping. We mapped the key stakeholders based on their expertise and involvement and validated it to ensure accuracy and relevance. Figure 4 visually represents the influence and interest levels of the project stakeholders, showcasing how each influence or is influenced by the EMS migration. This stakeholder map categorizes stakeholders based on their ability to influence the project outcomes and their interest in the project's success.

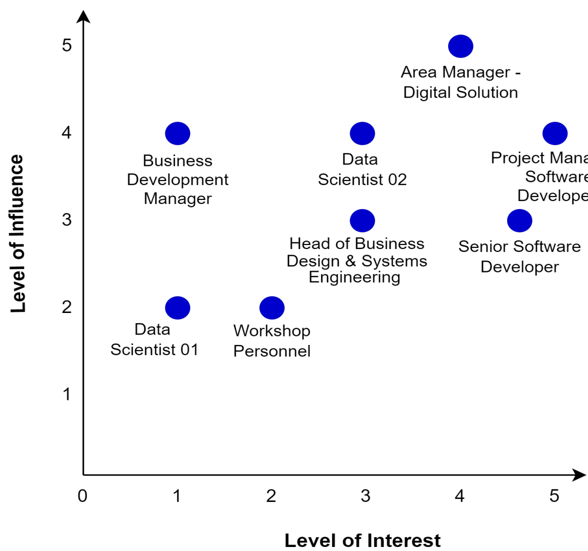

Figure 4. Stakeholder Mapping

Stakeholder Needs. We conducted the stakeholder analysis based on data from structured interviews. Table 4 summarizes the roles that each stakeholder plays in the migration process and their needs.

Table 4. Key Stakeholder Roles and Needs

Stakeholder	Role in Migration	Needs
Data Scientist 01	Developed predictive algorithms, important in initial planning	Clear documentation and guidance on optimizing algorithms on AWS
Data Scientist 02	Creates predictive tools, key for ongoing analytics	Tools to improve prediction time and reduce costs on AWS; training on AWS services
Senior Software Developer	Leads software coding, necessary for system function and scalability	Training on AWS technologies; advice on architectural changes for smooth integration
Project Manager	Keeps project on track, ensures alignment with goals	Clear communication; tools for tracking progress, identifying risks, and managing stakeholder expectations
Business Development Manager	Guides the market-oriented Energy Dashboard, drives commercial success	Proof of cost-effectiveness and market competitiveness; insights into pricing and market strategy
Head of Business Design and Systems Engineering Expert Group	Maintains technical and business standards, key for integrating innovative practices	Assurance that the migration plan aligns with technical and business standards; support for innovation and strategy
Area Manager – Digital Solution	Oversees digital strategy, important for aligning migration with digital plans	Assurance that the migration fits into broader digital strategies; insights into synergies with other projects
Workshop Personnel	Provides practical insights from the workshop, important for addressing operational issues	Training for any changes in operations; input on how migration could affect efficiency

Communication Plan. Table 5 outlines the strategies and tools used to engage the project stakeholders throughout the migration process. Such a stakeholder communication plan incorporates communication methods tailored to each stakeholder's roles and responsibilities, ensuring clear and effective information exchange.

Table 5. Stakeholder Communication Plan

Stakeholder	Method	Frequency	Key Messages	Feedback Mechanism
Data Scientist 01	Jira, Email, Zoom & In-Person Meetings	Bi-weekly	Updates on analytics tools, data handling	Feedback, Suggestion, Discussions, Emails, Team meetings
Data Scientist 02	Jira, Email, Zoom & In-Person Meetings	Bi-weekly	Updates on analytics tools, data handling	Feedback, Suggestion, Discussion, Emails, Team meetings
Software Developer	Jira, Email, Zoom & In-Person Meetings	Monthly	Architectural changes, coding practices	Feedback, Suggestion, Discussions, Emails, Team meetings
Project Manager	Jira, Email, Zoom & In-Person Meetings	Weekly	Project status, risk management updates	Feedback, Suggestion, Discussions, Emails, Team meetings, Review meetings
Business Development Manager	Presentations, Email, Zoom & In-Person Meetings	As needed	Market strategies, commercial impacts	Suggestion, Emails, and Project review meetings
Head of Business Design and Systems Engineering Expert Group	Presentations, Email, Zoom & In-Person Meetings	Quarterly	Technical standards, integration strategies	Feedback, Suggestion, Discussions, Emails, Project review meetings
Area Manager - Digital Solution	Presentations, Email, Zoom & In-Person Meetings	Weekly	Digital solution strategies, alignment with other projects	Feedback, Suggestion, Discussions, Emails, Team meetings, and Project review meetings
Workshop Personnel	Email, Zoom & In-Person Meetings	Pre and Post Migration	Operational changes, system usage tips	Feedback, Discussions, Emails, Team meetings

The Current Situation Analysis

Migration Strategy Framework. Figure 5 outlines a structured approach to migrating the EMS platform from StackHero to AWS. This process starts with careful planning by defining the migration's scope and objectives, aligning with the Company's strategic goals and stakeholder needs. The assessment phase encompasses a system analysis and risk assessment to identify and address potential challenges. In the strategy selection phase, the project stakeholders select the appropriate migration approach and choose the necessary tools for migration. This phase sets the foundational strategy for the actual migration process. During the execution phase, the migration plan, including application migration and pilot testing, will be implemented to ensure the migration is effective and meets expectations. After the migration, the focus will shift to validating system functionality and performance to achieve operational standards.

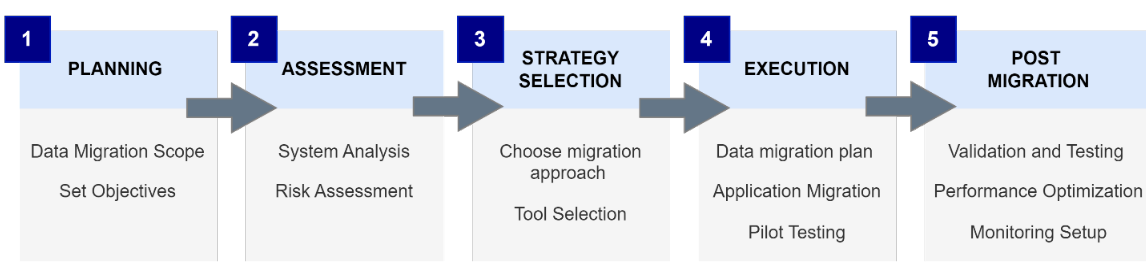

Figure 5. Migration Strategy Framework

AS-IS System Architecture. The Company's AS-IS system architecture features a Node-RED instance

on StackHero, acting as the central hub for automation and data integration, as shown in Figure 6. Sensors connect to Homey Pro, a smart home system, relaying data through a Message Queuing Telemetry Transport (MQTT) broker optimized for lightweight messaging with minimal bandwidth. Data stored in InfluxDB, a time-series database on StackHero, and a Lambda function in AWS processes it further to import e-Gauge energy meter data. This architecture integrates additional data streams like weather information from YR.no and designs for seamless data flow, real-time monitoring, and scalability. Adapting to evolving technological trends and harnessing improved operational efficiencies drives the shift from the AS IS to the TO-BE system architecture, which utilizes advanced AWS services.

Key essential functionalities identified in preparation for this migration include:

- Data Collection: Gathering diverse data streams from devices like e-Gauge energy meters.
- Real-time Processing: Enabling rapid decision-making through timely data analysis.
- Device Management: Ensuring smooth communication by integrating and managing multiple devices.
- Scalability: Handling increased data throughput without performance degradation.
- Reliability: Maintaining system availability is essential for real-time operations.
- Security: Securing data integrity and confidentiality across communications.
- Integration: Connecting effortlessly to other platforms and services.

These components are important for a robust, scalable, and secure EMS that is assured of future expansion, shown in Figure 6.

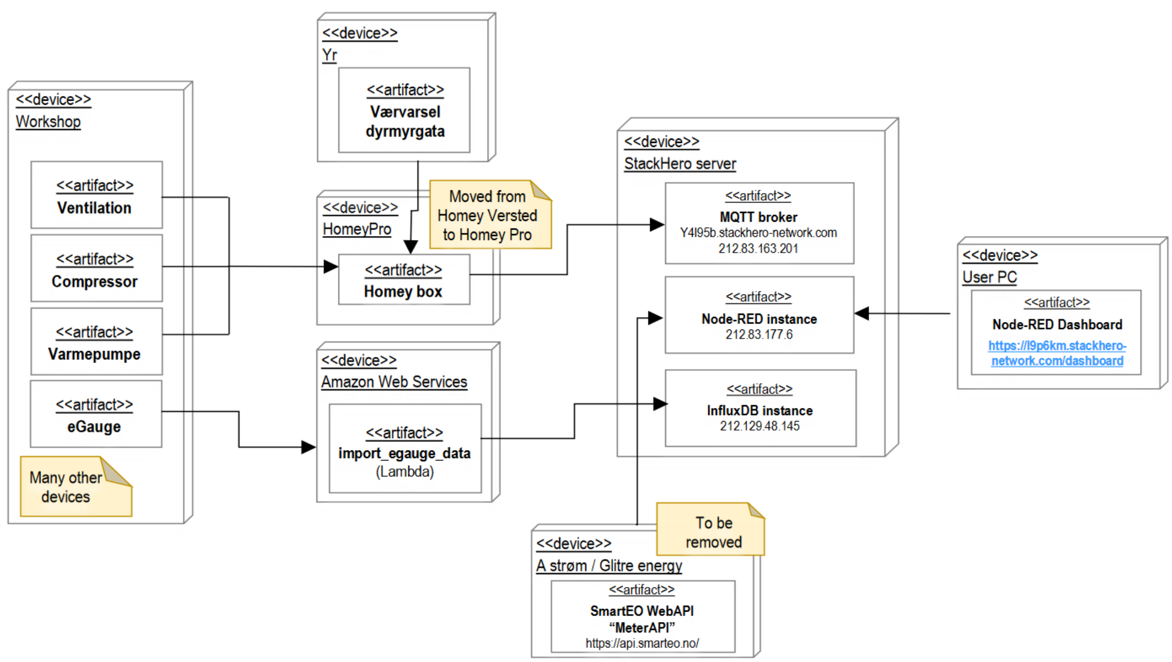

Figure 6. AS-IS System Architecture

MQTT Broker to AWS IoT Core. In the current setup, the MQTT Broker on StackHero is the central messaging hub for IoT systems, providing basic capabilities for real-time communication among devices. However, this system faces some limitations, such as limited scalability, requiring manual adjustments for higher loads; basic security measures, which may not suffice for protecting sensitive data; and simple real-time processing capabilities. In response, migrating MQTT Broker to AWS IoT Core is proposed as a solution. AWS IoT Core automatically handles millions of devices, offers robust end-to-end encryption, and features an advanced real-time rules engine for automated actions, enhancing responsiveness and security. It also integrates seamlessly with other AWS services, providing comprehensive device management. Table 6 shows a detailed comparison of the capabilities offered by AWS IoT Core over the MQTT Broker on StackHero.

Table 6. Comparison of MQTT Broker and AWS IoT Core

Feature	MQTT Broker on StackHero	AWS IoT Core
Data Collection	Relies on limited MQTT service capabilities.	Directly supports MQTT with large-scale data streams.
Real-time Processing	Basic or limited real-time processing capabilities.	Includes a real-time rules engine for automated actions.
Device Management	Basic or unclear device management.	Offers Device Gateway and Device Management features.
Scalability	Requires configuration; scalability limited.	Scales automatically to millions of devices.
Reliability	Basic reliability measures, likely less robust.	Built-in high availability and failover measures.
Security	Basic security: advanced features may be lacking.	Advanced security with end-to-end encryption.
Integration	Supports integration with possible configuration.	Seamless integration with other AWS services.

The decision to migrate from MQTT Broker to AWS IoT Core is driven by the company's need to strengthen its current performance by adopting several key system engineering practices. AWS IoT Core provides automatic scalability to handle a large number (millions) of devices without manual intervention, making it better suited for meeting the needs of growing IoT infrastructure. Higher security is another beneficial factor of IoT Core, which also brings along real-time processing capability, making it suitable for more complex applications such as energy mapping.

Node-RED instances on the AWS platform. A Node-RED instance on StackHero serves as the central hub for the EMS, facilitating data integration and real-time communication through connections to sensors and an MQTT broker. While functional, this system encounters significant scalability and security limitations that hinder its capacity to manage more significant, complex IoT deployments effectively. To overcome these challenges, we propose integrating Node-RED instance to AWS, allowing Node-RED to run in AWS, which promises enhanced scalability through automated management, improved security with IAM (Identity and Access Management) roles and policies, and better integration with other cloud services. This move promises enhanced scalability through automated management, improved security with IAM roles and policies, and better integration with other cloud services, shown in Table 7.

Table 7. Comparison of Node-RED instances on StackHero to Node-RED instances on AWS

Feature	Node-RED on StackHero	Node-RED on AWS
Data Collection	Provides standard Node-RED capabilities; integration with third-party services might require manual setup.	Extensive data collection capabilities, especially when integrated with AWS IoT and other AWS services.
Real-time Processing	Standard Node-RED functionality with additional plugins available for enhanced processing.	Powerful real-time processing when used with AWS services like AWS Lambda and Kinesis.
Device Management	Basic device management through Node-RED; may need external tools for complex scenarios.	Advanced device management capabilities through integration with AWS IoT Core.
Scalability	Manually scalable; dependent on the underlying server's resources provided by StackHero.	Highly scalable, ability to automatically adjust resources based on demand via AWS Auto Scaling.
Reliability	Reliability depends on StackHero's infrastructure; options for redundancy may be limited.	High reliability with multiple availability zones and backup options through AWS.

Security	Provides essential security features; additional configurations are needed for enhanced security.	Robust security features, including network firewalls, encryption, and AWS Identity and Access Management.
Integration	Supports integrations with various services, but might require more manual setup compared to AWS.	Seamless integration with a wide range of AWS services and third-party applications.

InfluxDB to AWS Timestream. The EMS project relies on InfluxDB as its primary database for fast data storage and querying in IoT and monitoring applications. However, using InfluxDB has presented challenges such as limited scalability under high-throughput demands and complex integration with additional cloud services, which can restrict the EMS's operational efficiency and adaptability. The introduction of Amazon Timestream for InfluxDB offers a solution to these challenges for the case study project. This enhancement provides a fully managed time-series database environment, incorporating features like automated scaling to accommodate varying loads, built-in high availability across multiple availability zones, and seamless integration with AWS's extensive cloud service suite. These capabilities address InfluxDB's limitations and strengthen security with AWS standards, including using Key Management Service (KMS) and Virtual Private Cloud (VPC) (shown in Table 8).

Table 8. Comparison of InfluxDB and AWS Timestream

Feature	StackHero InfluxDB	AWS Timestream for InfluxDB
Data Collection	Dependent on StackHero's service capabilities.	Supports InfluxDB API for direct integration.
Real-time Processing	Dependent on StackHero's service capabilities.	Enhanced with AWS infrastructure for faster processing.
Device Management	Likely provides some level of device management.	Integrated management through AWS's ecosystem.
Scalability	Should offer scalability options.	Designed to automatically scale with AWS capabilities.
Reliability	Includes reliability measures.	Built-in high availability across multiple AZs.
Security	Provides security measures.	Advanced security with AWS standards, including KMS and VPC.
Integration	May require additional configuration for integration.	Seamless integration with AWS services and other data tools.

Node-RED Dashboard to Amazon QuickSight. The Node-RED Dashboard tool is utilized to visualize real-time data in the EMS. Although it integrates well with various data sources, it needs to improve its scalability due to manual management and server dependence, which impacts larger implementations. Reliability and security also require extensive manual setup. Amazon QuickSight offers unique advantages for the EMS project with its comprehensive data analysis capabilities, seamless AWS integration, and fast processing supported by the (Super-fast, Parallel, In-memory Calculation Engine) SPICE engine. Its scalability and advanced security features ensure reliability, making it a better choice over Node-RED Dashboard (Table 9).

Table 9. Comparison of Node-RED Dashboard and Amazon QuickSight

Feature	Node-RED Dashboard	Amazon QuickSight
Data Collection	Supports integration with various data sources through flows and nodes.	Directly integrates with multiple AWS data sources and other external sources.
Real-time Processing	Capable of processing data in real time through lightweight nodes.	Offers real-time analysis with SPICE engine, enhancing data retrieval and manipulation speed.

Visualization Capabilities	Provides a customizable interface to build user-specific dashboards.	Offers a wide range of visualization types with deep customization options and automatic insights.
Scalability	Scalability is manually managed and depends on the underlying server.	Highly scalable, managed service that handles large-scale data workloads effortlessly.
Reliability	Reliability varies and is dependent on user-managed infrastructure.	High reliability with AWS infrastructure, ensuring data availability and system stability.
Security	Basic security features that require manual configuration for advanced needs.	Advanced security features, including row-level security, IAM integrations, and encryption at rest and in transit.
Integration	Flexible integration with IoT devices and third-party services through custom nodes.	Seamless integration with AWS services and existing BI (Business Intelligence) tools, enabling broader data strategy alignment.

TO-BE System Architecture

The proposed TO-BE systems architecture is illustrated in Figure 7, which was a collaborative effort with key stakeholders. We presented the initial version of the TO-BE system architecture to stakeholders for feedback. Based on the feedback and updates from Amazon and continuous stakeholder engagement, it evolved significantly. The technological rationale for the proposed architecture is based on detailed technological analyses with stakeholders in the following sections: MQTT Broker to AWS IoT Core, Node-RED Instances to the AWS platform, InfluxDB to AWS Timestream, and Node-RED Dashboard to Amazon QuickSight. Such rationale is used to validate this proposal, ensuring that the final design is theoretically sound and practically viable. Following these technical analyses, we distribute a validation survey to stakeholders for throughout validation. As such, we ensure the TO-BE system architecture aligns with the Company's strategic goals for innovative and sustainable energy management and meets the operational needs of the real situation.

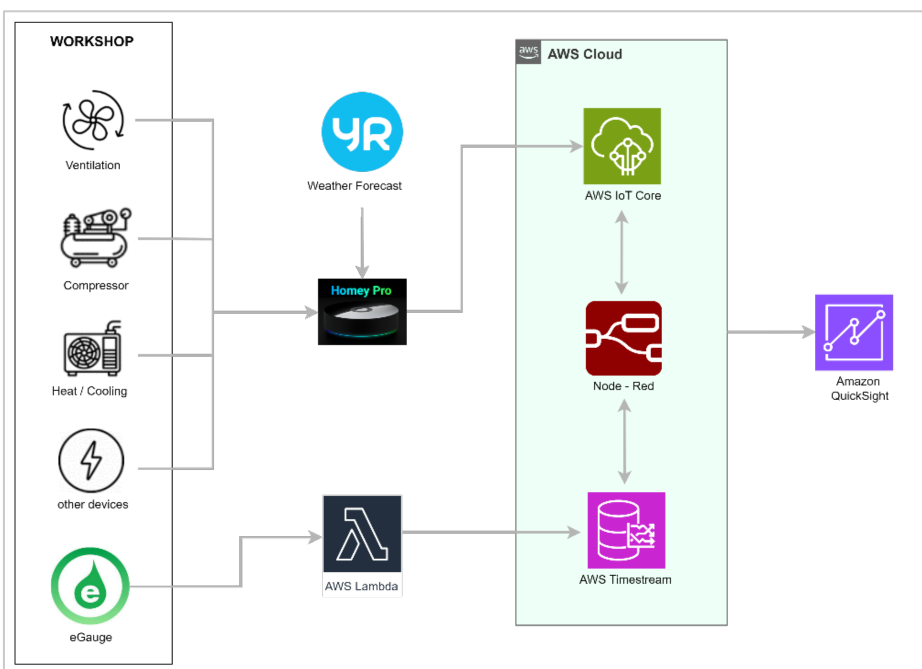

Figure 7. TO-BE System Architecture

Risks and Mitigation Strategies

Based on the interview data, we identified key risks associated with migrating the Company's platform from StackHero to AWS (shown in Figure 8). The five risks are elaborated below.

Technical Compatibility Issues. In the interview (SI 06), the project manager pointed to significant system architectural changes that conflict with existing applications or data formats. The project manager emphasized building a parallel AWS system alongside the existing Stackhero system, highlighting the need for careful integration to prevent compatibility issues.

Figure 8. Risk Identification

Data Security and Privacy Risks. Using IoT devices and handling sensitive data emphasized the need for robust security measures. In the structured interview (SI 02), there was a discussion about deploying algorithms on AWS and handling time series data, which highlighted concerns about data transfers and storage security.

Operational Downtime. In the interview (SI 06), the project manager mentioned the risk of service disruption during migration. This underscores the importance of maintaining real-time data access throughout the migration process, highlighting the risk of operational downtime and the need for continuous system availability.

Stakeholder Resistance. Resistance to adopting the new system may stem from their unfamiliarity with AWS services. In the interview (SI 06), the project manager suggested apprehension about AWS's user-friendliness.

Cost Overruns. There are significant concerns about potential cost overruns due to technical issues, training needs, and extended integration times. As the senior software developer said in the interview (SI 02), "the main concern in the project, as it is a non-customer project, is to keep the AWS costs low. So, we always evaluate the need to use a new AWS thing against the cost."

The identified risks underline the complexities of the migration and the critical need for careful planning and stakeholder engagement to address these challenges effectively. Based on the contemporary risk management theories and the extant best practices, Figure 9 outlines the strategy to address the challenges of the migration process.

Robust Testing and Validation. Robust testing and validation mechanisms must be employed to address technical compatibility issues. It is advisable to utilize sandbox environments and conduct pilot testing to ensure seamless integration and functionality, following the practices outlined by Kolb et al. (2015), who emphasize the importance of iterative testing in cloud environments to mitigate technical issues.

Figure 9. Mitigation Strategies

Enhanced Security Protocols. Due to the significant risks posed by data security and privacy vulnerabilities, it is important to implement enhanced security measures. These measures include encryption, multi-factor authentication, and frequent security audits. Mukherjee (2019) recommends adapting robust security frameworks to protect sensitive data during cloud migrations, mainly when working with IoT devices.

Stakeholder Engagement and Training. Detailed communication plans and comprehensive training sessions are necessary. They are essential in minimizing resistance and maximizing acceptance of the new system. Fowley et al. (2017) asserted that effective stakeholder engagement and education are vital to managing change and ensuring smooth technology transitions.

Budget Management and Contingency Planning. Effective cost management and avoiding budget overruns necessitate careful budgeting and planning for unforeseen circumstances. As Orue-Echevarri Arrieta et al. (2011) emphasized strategic financial planning for IT projects and incorporating a budget review mechanism are prudent steps to manage and mitigate financial risks associated with technological migrations.

System Redundancy and Backup. Establishing system redundancy and backup solutions is essential to minimizing operational downtime. Kim et al. (2012) highlighted the effectiveness of redundancy in maintaining system availability and continuity, particularly in cloud migration scenarios, which is essential for ensuring service reliability during and after the migration.

Validation of the Proposed System Architecture

The proposed solution is validated through the survey data. Figure 10 shows the validation results. The stakeholders scored functionality at an average of 3.75 and scalability and performance each at 5.0, indicating robust capabilities. Security and integration with current systems each received a score of 3.75, suggesting improvement areas for better operational fit. We found cost-effectiveness and stakeholder engagement to be less satisfactory (each scoring 3.0), pointing to a need for more inclusive financial domain engagement in the project team. Other assessed aspects, system adaptability (3.75), reliability (4.0), and user-friendliness (4.0), received general approval with room for improvement. Compatibility with existing systems and infrastructure scored 4.5, while risk identification and mitigation strategies were notably effective at 4.75 out of 5.

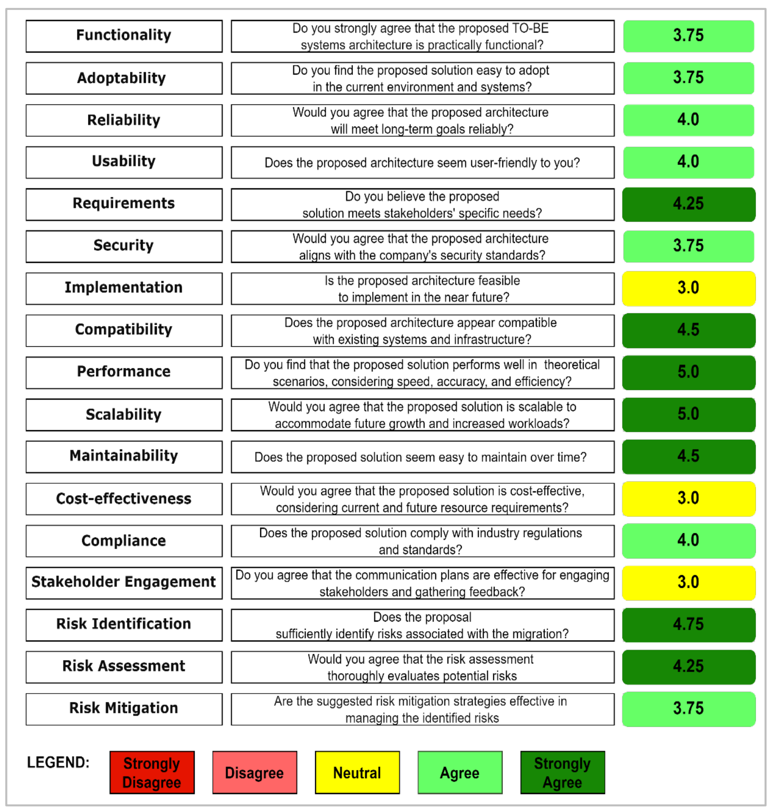

Figure 10. Survey Results

Discussion

The solution is a cloud-compatible system architecture, which replaced significant components with AWS-specific ones for a smooth migration. In the feedback workshop, the discussion of integrating AWS IoT Core, Amazon Timestream, Amazon QuickSight, and Node-RED provided initial explanations of the technical feasibility. The survey results further validated the system's functionality and scalability, aligning well with organizational goals to enhance operational effectiveness. It also reveals areas for improvement, highlighting concerns in cost and stakeholder engagement, etc. It's important to note that such a validation method may not capture the full nuance and complexity of stakeholder sentiments, particularly concerning new technological integrations, and might introduce biases in self-reported data prior to the implementation.

Nevertheless, the case study sheds light on the important role of stakeholder engagement and risk management in successful migration within complex environments. The methodology used in this study may be applied to redesign architectures required to integrate novel functionalities in other platform cases. This study contributes to the broader discourse on systems engineering methodology on platform migration. Specifically, designing system architecture for platform migrations requires integrating novel functionalities, improving efficiency without disrupting the service. These aspects need to be considered in all the project life cycle phases of the platform.

Future research could further verify the solution during the implementation of the system architecture and explore new technologies to enhance the integration and reliability of AWS architectures. The work could cover, for instance, developing verification techniques to assess AWS configurations' performance, security, and scalability, to maintain robustness against evolving demands; studying the long-term effects of system architectural changes on performance and user satisfaction for sustainable operation; and economic analyses of migrations in maintaining viability and optimizing cost-efficiency.

References

Agavanakis, K., Sakellarakis, K., & Koutroubinas, S. (2012). Moving Intelligent Energy applications upwards: A customer oriented cloud solution. *The 1st IEEE Global Conference on Consumer Electronics,* (pp. 607-611).

Amazon Timestream for InfluxDB is now generally available. (2024, March 15). Amazon Web Services, Inc. https://aws.amazon.com/about-aws/whats-new/2024/03/amazon-timestream-influxdb-available/

Amazon Web Services, Inc. (2024). Amazon Web Services. https://aws.amazon.com/

Binz, T., Breitenbücher, U., Kopp, O., & Leymann, F. (2014). *Migration of enterprise applications to the cloud. It - Information Technology, 56(3).*

Dabbagh, M., Hamdaoui, B., Guizani, M., & Rayes, A. (2018). An Energy-Efficient VM Prediction and Migration Framework for Overcommitted Clouds. *IEEE Transactions on Cloud Computing*, 6, 955-966.

Engelsrud, A. (2019). Moving to the Cloud: Lift and Shift. *Managing PeopleSoft on the Oracle Cloud*.

Fowley, F., Elango, D., Magar, H., & Pahl, C. (2017). Software System Migration to Cloud-Native Architectures for SME-Sized Software Vendors. In: Steffen, B., Baier, C., van den Brand, M., Eder, J., Hinchey, M., Margaria, T. (eds) SOFSEM 2017: Theory and Practice of Computer Science. SOFSEM 2017. Lecture Notes in Computer Science, vol 10139. Springer, Cham. (pp. 498-509).

Gaza, C.K.M.T., Giudici, H., Falk, K. (2024). Enhancing Industrial Energy Management: Improving Efficiency and Stakeholder Satisfaction. *The Proceedings of the 2024 Conference on Systems Engineering Research Conference on Systems Engineering Research Series* (pp. 85-101).

Gupta, K., & Katiyar, V. (2016). Energy aware virtual machine migration techniques for cloud environment. *International Journal of Computer Applications*, 141(2), 11-16.
Hilburn, T. (2023). Software engineering features - Models, methods, tools, standards, and metrics. In INCOSE, *Systems Engineering Handbook: Guide to the Systems Engineering Body of Knowledge (SEBoK)* (Version 2.9, pp. 956-959).
INCOSE (2023) *Systems Engineering Handbook: Guide to the Systems Engineering Body of Knowledge* (SEBoK) (Version 2.9, November 20, 2023).
Khajeh-Hosseini, A., Greenwood, D., & Sommerville, I. (2010). Cloud migration: A case study of migrating an enterprise IT system to IaaS. *IEEE 3rd International Conference on Cloud Computing,* (pp. 450-457)
Kim, W., Lee, J. H., Hong, C., Han, C., Lee, H., & Jang, B. (2012). An innovative method for data and software integration in SaaS. *Computers & Mathematics with Applications,* 64(5), 1252–1258.
Kolb, S., Lenhard, J., & Wirtz, G. (2015). Application migration effort in the cloud. *Services Transactions on Cloud Computing*, 3(4), 1-15.
Mukherjee, S. (2019). Benefits of AWS in modern cloud. *SSRN Electronic Journal*.
Muller, G., Giudici, H. (2024). Social Systems of Systems Thinking to Improve Decision-Making Processes Towards the Sustainable Transition. *The Proceedings of the 2024 Conference on Systems Engineering Research,* (pp. 341-353).
Mwansa, G., & Mnkandla, E. (2014). Migrating Agile Development into the Cloud Computing Environment. *IEEE 7th International Conference on Cloud Computing* (pp. 818-825).
Olariu, F. (2023). Overcoming Challenges in Migrating Modular Monolith from On-Premises to AWS Cloud. *22nd RoEduNet Conference: Networking in Education and Research (RoEduNet)*, 1-6.
Orue-Echevarria Arrieta, L., Alonso Ibarra, J., Gottschick, J., & Restel, H. (2011). *From software-as-a-good to SaaS: Challenges and needs - Developing a tool-supported methodology for the migration of non-SaaS applications to SaaS*. In M. J. Escalona Cuaresma, B. Shishkov, & J. Cordeiro (Eds.), ICSOFT 2011 - Proceedings of the 6th International Conference on Software and Data Technologies, Vol. 2 (pp. 257-260). SciTePress.
Pahl, C., & Xiong, H. (2013). Migration to PaaS clouds - Migration process and architectural concerns. *IEEE 7th International Symposium on the Maintenance and Evolution of Service-Oriented and Cloud-Based Systems*.
Pineda, R. (2023). Product architecture, modeling, and analysis. In INCOSE, *Systems Engineering Handbook: Guide to the Systems Engineering Body of Knowledge (SEBoK)* (Version 2.9, pp. 595-599).
Reguri, V., Kogatam, S., & Moh, M. (2016). Energy Efficient Traffic-Aware Virtual Machine Migration in Green Cloud Data Centers. *2016 IEEE 2nd International Conference on Big Data Security on Cloud (BigDataSecurity), IEEE International Conference on High Performance and Smart Computing (HPSC), and IEEE International Conference on Intelligent Data and Security (IDS),* (pp. 268-273).
Running on Amazon web services. (n.d.). Node-RED. https://nodered.org/docs/getting-started/aws
Samrajesh, M., & Gopalan, N. (2013). Component-based energy-aware multi-tenant application in software as a service*. 15th International Conference on Advanced Computing Technologies (ICACT)*, (pp. 1-5).
Semcon. (2024). Official website. Semcon. https://www.semcon.com/
Shastry, A., Nair, D., Prathima, B., Ramya, C., & Hallymysore, P. (2022). Approaches for migrating non cloud-native applications to the cloud. *IEEE 12th Annual Computing and Communication Workshop and Conference (CCWC),* (pp. 0632-0638).
Son, A.-Y., Byun, J. Y., Yong, C., Huh, E.-N., Hyun, J. H., & Kang, K. K. (2017). *Energy efficiency oriented migration scheme in cloud data center. IEEE International Conference on Cybernetics and Computational Intelligence (CyberneticsCom)*, (pp.122-127).
Yang, C., Chen, S., Liu, J., Liu, R., & Chang, C. (2019). On construction of an energy monitoring service using big data technology for the smart campus. *Cluster Computing*, 23, 265-288.

Yang, C., & Wan, T. (2020). Implementation of an energy-saving cloud infrastructure with virtual machine power usage monitoring and live migration on OpenStack. *Computing*, 102, 1547-1566.
Zheng, J., & Du, W. (2014). Toward easy migration of client-server applications to the cloud. *9th International Conference on Software Engineering and Applications,* (pp.101-108).

Abbreviations

ABD: Architectural Block Diagrams

API: Application Programming Interface

AWS: Amazon Web Services

AZ: Availability Zone

BI: Business Intelligence

DevOps: Development and Operations

EMS: Energy Monitoring System

FBD: Functional Block Diagrams

IAM: Identity and Access Management

IoT: Internet of Things

ISO/IEC: International Organization for Standardization/ International Electrotechnical Commission

KMS: Key Management Service

MQTT: Message Queuing Telemetry Transport

SEBoK: Guide to the Systems Engineering Body of Knowledge

SLA: Service-Level Agreements

SPICE: Super-fast, Parallel, In-memory Calculation Engine

SQS: Simple Queue Service

S3: Simple Storage Service

VM: Virtual Machine

VPC: Virtual Private Cloud

Biography

Catalina Klarissa Mae Tagavilla-Gaza. Klarissa has a background in civil engineering and business administration and holds a master's degree in innovation and technology management specializing in systems engineering from the University of South-Eastern Norway. She previously worked in the Philippines construction industry before moving to Norway to pursue her BBA. Currently, she works as a project planner for a subsea oil and gas company in Norway.

Henri Giudici. Henri Obtained his Ph.D. from the Norwegian University of Science and Technology (NTNU) in 2019. After his Ph.D., Henri worked for three years in the R&D department of a Norwegian company dealing with smart road monitoring systems. Currently Henri holds a position at the University of South Eastern Norway (USN) as a Researcher in Systems Engineering and Innovation.

YangYang Zhao. Dr. Zhao is currently an Associate Professor in Entrepreneurship and Digitalization at the University of Oslo and in Systems Engineering at the University of South-Eastern Norway. Her recent research focuses on business process reengineering, fuzzy-front end design, AI and creative (team)work, responsible digital transformation, etc. She has extensive interdisciplinary research and teaching experience across Europe, the US, and Asia. Additionally, she has an industrial background in new venture creation within the IT sector. Her scholarly work has been published in numerous peer-reviewed journals and conferences.