



Empirical studies of power consumption in identity management systems

Lappeenranta-Lahti University of Technology LUT

LUT School of Engineering Science

Software Engineering

2023

Micky Yun Chan

Examiners: Professor Jari Porras

Professor Patricia Lago

Professor Henry Muccini



SE4GD
Software Engineers for Green Deal



With the support of the
Erasmus+ Programme
of the European Union

This thesis has been accepted by partner institutions of the consortium (619839-EPP-1-2020-1-FI-EPPKA1-JMD-MOB).

Successful defence of this thesis is obligatory for graduation with the following national diplomas:

- Master of Computer Science (University of L'Aquila)
- Master of Science in Technology (LUT University)
- Master of Computer Science (Vrije Universiteit Amsterdam)

ABSTRACT

Lappeenranta-Lahti University of Technology LUT

LUT School of Engineering Science

Software Engineering

Micky Yun Chan

Empirical studies of power consumption in identity management systems

Master's thesis

2023

63 pages, 17 figures, 33 tables and 1 appendices

Examiners: Professor, Jari Porras
Professor, Patricia Lago
Professor, Henry Muccini

Supervisor: Professor, Lau Sian Lun (Sunway University)

Keywords: Identity management, power consumption measurement, blockchain, empirical software engineering

The increasing interest in blockchain-based decentralized identity solutions has raised concerns regarding their energy consumption and impact on sustainability. This study addresses these concerns by conducting experiments to measure power consumption in three identity solutions based on centralized, federated, and decentralized identity models and technologies. The experiments were conducted in a controlled environment utilizing Docker containers to ensure a clean and standardized testing environment. Power consumption data was collected, and statistical analysis, including the analysis of variance (ANOVA) and Tukey's honest significant difference (HSD) test, was performed to compare the power consumption across the different identity models. The findings indicate that blockchain-based decentralized identity solutions do not necessarily consume more power than other identity solutions. This research provides valuable insights into the energy efficiency of blockchain-based decentralized identity solutions, contributing to the ongoing discourse on sustainable technologies in the field of identity management.

ACKNOWLEDGMENTS

I would like to express my heartfelt gratitude to Prof. Lau Sian Lun from Sunway University, Malaysia, for his exceptional supervision and guidance throughout the journey of my thesis. His willingness to answer my numerous questions and provide invaluable advice has been instrumental in shaping the outcome of this research and has extended beyond academia to positively impact various aspects of my life.

Furthermore, I am grateful to Prof. Jari Porras for organizing the Erasmus Mundus SE4GD program and to all the Professors and staff associated with the program. Participating in the SE4GD program has been a transformative experience, offering unparalleled learning opportunities and broadening my horizons in the field of software engineering for sustainable development.

I would like to extend a special note of thanks to my parents for their unwavering support and belief in my choices and aspirations. Their encouragement has been a constant source of strength, motivating me to persist and excel in my endeavors.

Lastly, I take a moment to thank myself for daring to step out of my comfort zone and embrace challenges. This journey of growth and self-discovery would not have been possible without the courage to pursue new paths and face uncertainties.

To everyone who has contributed to my academic and personal growth, I am truly grateful for your presence and support. Your influence has played an integral role in shaping the person I am today, and for that, I extend my appreciation.

ABBREVIATIONS

Abbreviations

EBSI European Blockchain Services Infrastructure

EU European Union

IdM Identity management

IdP Identity Provider

IT Information technology

PoA proof-of-authority

PoW proof-of-Work

SP Service Provider

UN United Nation

TABLE OF CONTENTS

ABSTRACT	iii
ACKNOWLEDGMENTS	iv
SYMBOLS AND ABBREVIATIONS	v
1 INTRODUCTION	5
1.1 Background and Motivation	5
1.2 Research Objective and Scope	6
1.3 Research Method	7
1.4 Related Work	8
1.4.1 Power/Energy consumption in Blockchain	8
1.4.2 Power/Energy measurement methodology for software	8
1.5 Structure of Report	9
2 BACKGROUND KNOWLEDGE	10
2.1 Identity management system	10
2.1.1 Concept of Identity	10
2.1.2 Conceptual Components in Identity management system	10
2.1.3 Operations of Identity management system	12
2.2 Type of Identity management system model	13
2.2.1 Centralized Model	13
2.2.2 Federated Model	15
2.2.3 Decentralized Model	17
2.3 Software defined power measurement	19
2.4 Comparison between Identity management system models	19
3 METHODOLOGY	21
3.1 Methodology Overview	21
3.2 Experimental Design	22
3.2.1 Research Variables	22
3.2.2 Experiment Environment	22
3.2.3 Experimental Run Design	22
3.2.4 Hypothesis Formulation	23
3.3 Identity system setup	24
3.3.1 Centralized identity system	24
3.3.2 Federated identity system	25

3.3.3	Decentralized identity system	26
3.4	Summary of Identity system	27
4	RESULT	28
4.1	Collected Data	28
4.1.1	Power Consumption for Register	29
4.1.2	Power Consumption for Login	30
4.1.3	Power consumption for processes	31
4.2	Statistical Test Analysis	34
4.2.1	Statistical test for RQ1	34
4.2.2	Statistical test for RQ2	38
4.2.3	Statistical test for RQ3	44
5	CONCLUSION	53
5.1	Summary of Findings	53
5.1.1	RQ1: Difference in power consumption between different model of identity systems	53
5.1.2	RQ2: Power consumption of different type of identity management system vary with different levels of user activity	53
5.1.3	RQ3: Components that consume the most power in identity manage- ment system	54
5.2	Discussion of Result	54
5.2.1	contribution of the study	54
5.2.2	Limitation of the experiment	55
5.2.3	Future work	55
	REFERENCES	56
	APPENDICES	
	A APPENDIX: SOURCE CODE	
	LIST OF FIGURES	
1	Example of Identity	10
2	UML Sequence Diagram for Registration	12
3	UML Sequence Diagram for Authentication (LOGIN)	12
4	Centralized Model	14
5	Federated Model	15
6	Decentralized Model	17

7	Flow Chart for Methodology	21
8	Power Consumption for Register Operations	29
9	Power Consumption for Register 20/second Operation	29
10	Power Consumption for Login Operation	30
11	Power Consumption for Login 20/second Operation	30
12	Power Consumption for Register operation in processes	31
13	Power Consumption for Login operation in processes	31
14	Power Consumption for Register operation in processes	32
15	Power Consumption for Login operation in processes	32
16	Power Consumption for Register operation in processes	33
17	Power Consumption for Login operation in processes	33

LIST OF TABLES

1	Research Questions	7
2	Comparison of Identity Management Models	19
3	Research Variables	22
4	Parameters of the experiment run	23
5	Hypotheses for Research Questions	24
6	Implementation Details of Identity Solutions	27
7	Components of Identity Solutions	27
8	ANOVA test for Register Operation between identity systems	34
9	ANOVA test for Login Operation between identity systems	34
10	Tukey's HSD Pairwise Group Comparisons for 5 register/sec	36
11	Tukey's HSD Pairwise Group Comparisons for 10 register/sec	36
12	Tukey's HSD Pairwise Group Comparisons for 20 register/sec	36
13	Tukey's HSD Pairwise Group Comparisons for 50 register/sec	37
14	ANOVA test for register operation at different send rate (5,10,20,50)	38
15	ANOVA test for login operation at different send rate (5,10,20,50)	38
16	Tukey's HSD Pairwise Group Comparisons for Centralized system (Register)	40
17	Tukey's HSD Pairwise Group Comparisons for Centralized system (Login)	40
18	Tukey's HSD Pairwise Group Comparisons for Decentralized system (Register)	41
19	Tukey's HSD Pairwise Group Comparisons for Decentralized system (login)	41
20	Tukey's HSD Pairwise Group Comparisons for Federated system (Register)	42
21	Tukey's HSD Pairwise Group Comparisons for Federated system (Login)	43
22	ANOVA test for register processes of centralized Identity system	44

23	ANOVA test for login processes of centralized Identity system	44
24	ANOVA test for register in processes of decentralized Identity system	45
25	ANOVA test for login in processes of decentralized Identity system	45
26	ANOVA test for login in processes of federated Identity system	46
27	ANOVA test for register in processes of federated Identity system	46
28	Tukey's HSD Pairwise Comparisons for Processes in Centralized system (Register)	48
29	Tukey's HSD Pairwise Comparisons for Processes in Centralized system (Login)	48
30	Tukey's HSD Pairwise Comparisons for Processes in Decentralized system (Register)	49
31	Tukey's HSD Pairwise Comparisons for Processes in Decentralized system (Login)	49
32	Tukey's HSD Pairwise Comparisons for Processes in Federated system (Register)	51
33	Tukey's HSD Pairwise Comparisons for Processes in Federated system (Login)	52

1 INTRODUCTION

1.1 Background and Motivation

Identity management has always been an issue across different organizations with known exploitation techniques such as social engineering, data leaking etc. Recent compromises in the identity management system have caused great economic losses and social impact due to private personal data leak to public. One study (Lim et al. 2018) had proposed that decentralized identity systems could prevent some of these exploitation. Identity forgery is a prevalent issue across industries such as financial, health, and the Information technology (IT) industry.

Decentralized identity model has promised several improvements compared to centralized and federated approaches. It can reduce the time required to verify an individual, minimize risks for data leak prevalent in a centralized approach, and easily detect unauthorized identity alterations due to the nature of blockchain. However, the implementation with blockchain approaches often comes with several technological overheads, such as the cost of running multiple nodes with the same set of data and lower transaction time. Scepticism toward blockchain technologies has created obstacles in the adoption of blockchain. Despite this International organizations such as the European Union have established initiatives such as European Blockchain Services Infrastructure (EBSI)¹ to develop a data exchange system which utilize blockchain technologies across all European Union (EU) members with use cases such as social security, diploma, insurance, etc.

Sustainability is defined as “meeting the needs of the present without compromising the ability of future generations to meet their own needs”² by the United Nation (UN). It is not until recently that sustainability in software has seen a rise of interest in academic institutions and international organizations such as the UN and EU. A clear example of this is the funding of sustainable software Erasmus Mundus programs such as SE4GD³, PERCCOM(Ah-Lian et al. 2019), and GENIAL⁴ by the EU. These programs are designed to educate software engineers to optimize IT solution to be more energy efficient or to build IT solutions that help achieve sustainable development goals. One hard metric to evaluate sustainable software is its power consumption. Measuring software power consumption has been a challenge without the use of specific hardware. Various approaches to software-based power consumption have been

¹<https://ec.europa.eu/digital-building-blocks/wikis/display/EBSI/Node+Operators>

²<https://www.un.org/en/academic-impact/sustainability>

³<https://se4gd.lutsoftware.com/>

⁴<http://genial.univ-lorraine.fr/programme/>

proposed. One interesting approach is called Software-defined power meters, which has been developed to measure power consumption of software without any need for specialized hardware. It works using an empirical regression model designed from benchmarked data or using CPU built-in power meter for power consumption measurement. One such tool which has been developed recently is called PowerJoular (Noureddine 2022), which measures power consumption in Intel x86 architecture and single board computers such as the Raspberry Pi.

Power consumption in blockchain has been a major focus ever since the popularity of cryptocurrency came into fame, but since the majority of research focuses on public blockchain, also known as permissionless blockchain, it is important not to use generalized power consumption statements to all blockchain technologies as it is not a homogeneous group of technologies. For instance, the power consumption between permissioned and permissionless blockchain can have considerable differences. It is important to note that power consumption of blockchain varies differently depending on factors such as consensus algorithm, reward mechanism, and whether or not the blockchain is permissioned or permissionless. (Schinckus 2020)

There are an abundance of papers that evaluate power consumption of blockchain technologies, however, there is surprisingly little research done on the power consumption of decentralized identity systems. It could be partly due to the relatively new field of utilizing blockchain technologies for identity systems and the difficulties of setup. (Alshahrani et al. 2023)

Due to the rising interest in sustainability and decentralized Identity management (IdM), it is important to have data on the power consumption of such system compared to the alternatives, this in turn give an insight on the sustainability of such technologies.

1.2 Research Objective and Scope

The main objectives of this thesis are to provide an overview of decentralized identity systems power consumption, therefore a proof of concept comparison with existing identity system approaches with an emphasis on power efficiency. Comparison in terms of power consumption used will be shown between the three identity systems in decentralized, federated, and centralized implementation. Furthermore, the power consumption data are systematically gathered from individual components within the systems. This enables a detailed analysis of the power consumption patterns of each component, offering valuable insights into the components that exhibit the highest power consumption. Consequently, these findings provide crucial guidance for optimizing the power consumption of the IdM systems. It is hoped that the evaluation of these systems will give an estimated picture of the amount of resources they consume.

With these objectives, three research questions are formalized as shown on Table 1.

Research question 1 is intended to investigate the difference in power consumption between decentralized, federated and centralized identity systems. The power consumption of identity management systems can have a significant impact in large-scale production deployments therefore it is necessary to have an insight on power consumption of such systems. Furthermore by having power consumption data, it could be ultimately used on sustainability and efficiency assessment.

Research question 2 is aimed to investigate the relationship between power consumption of identity management systems and different levels of user activity. This is necessary because there could be a scenario that where a particular type of identity management system could be more power efficient when there are higher load. It also provide an insight on which type of identity management system fit what purposes and allow software architect to choose most suitable type of system.

Research question 3 aims to investigating the power consumption of different type of components within identity management system. By doing this, it allow identification of components that consume most power and potentially allow optimization in power efficiency.

For all the research questions, a statistical approach will be employed, involving the formulation of hypotheses. These hypotheses will then be addressed using data collected from empirical experiments, and the analysis will be conducted through statistical tests.

Research Question	Objective
What is the difference in power consumption between traditional centralized identity management systems and decentralized identity management systems?	To determine Whether or not decentralized identity system consume more power than other alternatives identity system
How does the power consumption of different type of identity management system vary with different levels of user activity?	To investigate the impact of load on decentralized, federated and centralized identity system
What are the components that consume the most power in identity management system?	To identify the component within identity system that consume the most power

Table 1: Research Questions

1.3 Research Method

To answer Research questions, a review on the existing related technologies was presented to ensure a sufficient understanding. Then identity systems based on three IdM models are implemented either using the existing framework or from scratch. Experiments which utilized

software defined power measurement tool to collect power consumption data of the three IdM systems. From the collected data, statistical tests are performed and from the result we can inference the answer to the research questions.

1.4 Related Work

This section provides a discussion and summary of studies that share similarities with the present study.

1.4.1 Power/Energy consumption in Blockchain

According to (Sedlmeir et al. 2020), the difference in power consumption per transaction between a proof-of-Work (PoW) blockchain and a non-PoW public blockchain differs by over 100%. Furthermore, enterprise blockchains consume 50% less energy compared to non-PoW public blockchains. Based on the experimental results of this paper, it is concluded that enterprise blockchains consume more energy per transaction than a centralized system. However, we cannot immediately assume that similar results can be obtained from decentralized identity systems, which are often built upon enterprise blockchains. The power consumption of permissioned blockchains with proof-of-authority (PoA) is sensitive to the size of the network, and it is concluded that the consensus algorithm can have a significant impact on power consumption.

In (Stokkink et al. 2021), the performance of credential enrollment and verification of different decentralized IdM systems, including Hyperledger Indy, uPort, and Jolocom, was measured and compared. The study concluded that identity data disclosure protocols are not the only factor in an decentralized IdM system that causes a difference in performance, and from the empirical results, it is shown that different implementations of decentralized IdMs have noticeable differences in CPU usage.

1.4.2 Power/Energy measurement methodology for software

There are various way of measuring power consumption and energy efficiency in software system. The systematic literature review performed by (Ergasheva et al. 2020) listed metrics and tools that were used in power consumption measurement in over 500 papers. (Hindle 2012) developed an abstract Methodology of Relating Software Change to Power Consumption which consist of seven stages listed below:

- 1 : Choosing a product and a context
- 2 : Decide on measurement and instrumentation:
- 3 : Choose a set of versions

- 4 : Developing a test case
- 5 : Configure the testbed
- 6 : Per each version
- 7 : Compile and Analyze the results

(Acar et al. 2016) introduce a model that estimate the power consumed by CPU, memory and disk due to the execution of an application at runtime, in a way it can estimate the impact of source code on power consumption. This is achieved though having mathematical formula for power consumption in CPU, memory and disk.

1.5 Structure of Report

Chapter 1 - This chapter outline the background, motivation, objectives of the study. In Addition, a summary of research method and a paragraph on related work are provided.

Chapter 2 - In this chapter, background knowledge on the technologies and concept of software defined power measurement and identity management systems are provided.

Chapter 3 - In this chapter, detailed methodology of the experiment are provided. It include experimental design and implementation of the test subject which is the identity systems.

Chapter 4 - This chapter showcase the statistical test results from the experiments.

Chapter 5 - This chapter bridge the experimental result with the RQs and hypotheses and draw a conclusion for the study. In the end, a paragraphs on future work and study limitations are provided.

2 BACKGROUND KNOWLEDGE

In this chapter an introduction of background knowledge necessary to understand identity management systems and software defined power measurement in the context of the experiment setup are provided.

2.1 Identity management system

2.1.1 Concept of Identity

In order to understand identity management system, the concept of identity must first defined. Identity consist of the following three data: Identifiers, credentials and attributes. (Cao and Yang 2010) Identifiers are any data that can be used to identify a subject, for example account name, passport number etc. Credentials on the other hand are data that were used to verify one's identity i.e verification of one's identity. Attributes describes the characteristics of a subject, it could include details such as birthday, user rights etc. (Bertino and Takahashi 2010)

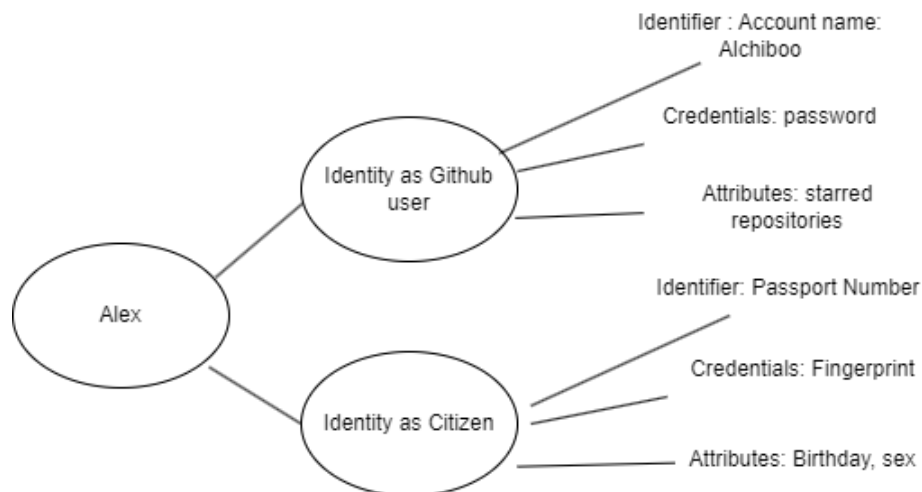


Figure 1: Example of Identity

2.1.2 Conceptual Components in Identity management system

In Identity management system, it includes the following components :

- User - Users represent an unique identity in the Identity management system. It is client to both Service Provider (SP) and Identity Provider (IdP).
- SP - Service provider provide services to users, for example it could be online shop.

- IdP - Identity provider is the core component of an IdM system. It provide primarily two functionalities to both users and SPs. The first function is registration which is a process that record user's identity into system storage. The second function is authentication, which is authenticate identity of users or SPs that request access to the system.

The implementation details of these components can be vary depend on technologies and IdM models.

Figure 1 presents an illustrative example of identity, where the user is named Alex. This example encompasses two distinct identities associated with Alex: a GitHub user identity and a citizen identity. The GitHub user identity is characterized by an account name as the identifier, a password as the credential, and attributes such as starred repositories. In contrast, the citizen identity utilizes a passport number as the identifier, a fingerprint as the credential, and attributes like birthday and sex.

2.1.3 Operations of Identity management system

According to the guideline established by (Grassi et al. 2020), the operations of IdM include enrollment aka registration, authentication and identity life cycle management.

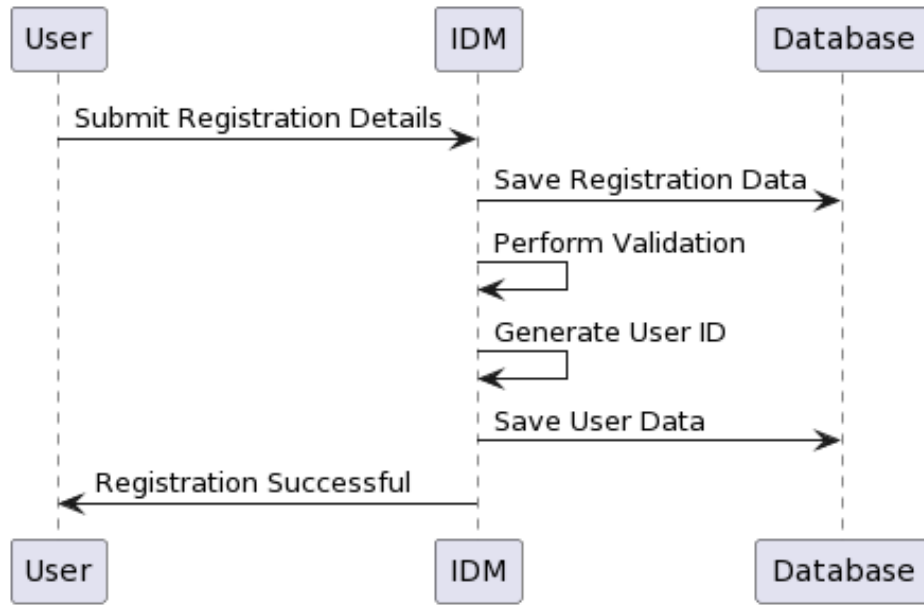


Figure 2: UML Sequence Diagram for Registration

Figure 2 represents the process of user registration in an Identity Management System. It involves three participants: User, IdM, and Database. The sequence of events is as follows: The User submits their registration details to the IdM. The IdM saves the registration data in the Database. The IdM performs validation on the registration data. The IdM generates a unique User ID for the new user. The IdM saves the user data, including the generated User ID, in the Database. The IdM notifies the User that the registration was successful. This diagram illustrates the steps involved in the user registration process, including data storage and validation, as well as the communication between the User, IdM, and Database participants.

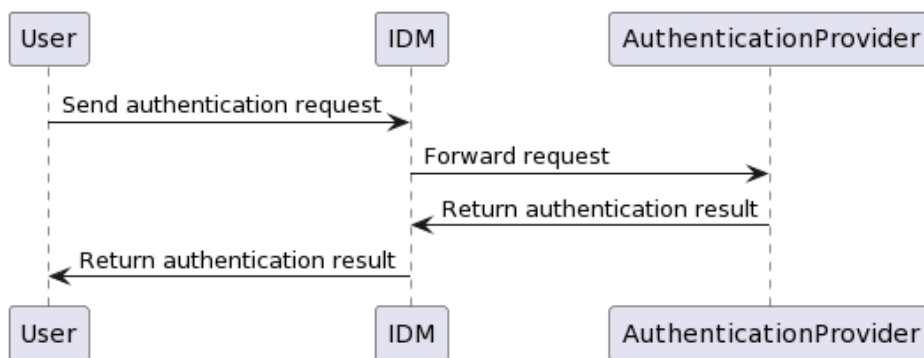


Figure 3: UML Sequence Diagram for Authentication (LOGIN)

Figure 3 represents the process of authentication in an Identity Management System (IdM). It involves three participants: User, IdM, and Authentication Provider. The sequence of events is as follows: The User sends an authentication request to the IdM. The IdM forwards the authentication request to the Authentication Provider. The Authentication Provider processes the request and returns the authentication result to the IdM. The IdM receives the authentication result from the Authentication Provider. The IdM then returns the authentication result to the User. This diagram illustrates the flow of communication and the order in which the participants interact during the authentication process.

2.2 Type of Identity management system model

The model of IdM illustrate how each conceptual components interact with each other. The model of IdM can be primarily classified into three types: centralized, federated and decentralized. It is important to noted that this section only distinguish the conceptual difference not the technological difference.

2.2.1 Centralized Model

Figure 4 depicts the centralized identity model, characterized by a single central identity provider. In this model, users send their requests to the service provider, which subsequently relays these requests to the central identity provider. It is worth noting that a single actor can simultaneously function as both the service provider and the identity provider. In figure ??, interaction between each components in the centralized model are illustrated.

The isolated model is a variant of centralized model where SP and IdP are combined into single entity and hence the term isolated model. This particular model is prone to system failure due to the over centralization of critical IdM functions. (Cao and Yang 2010)

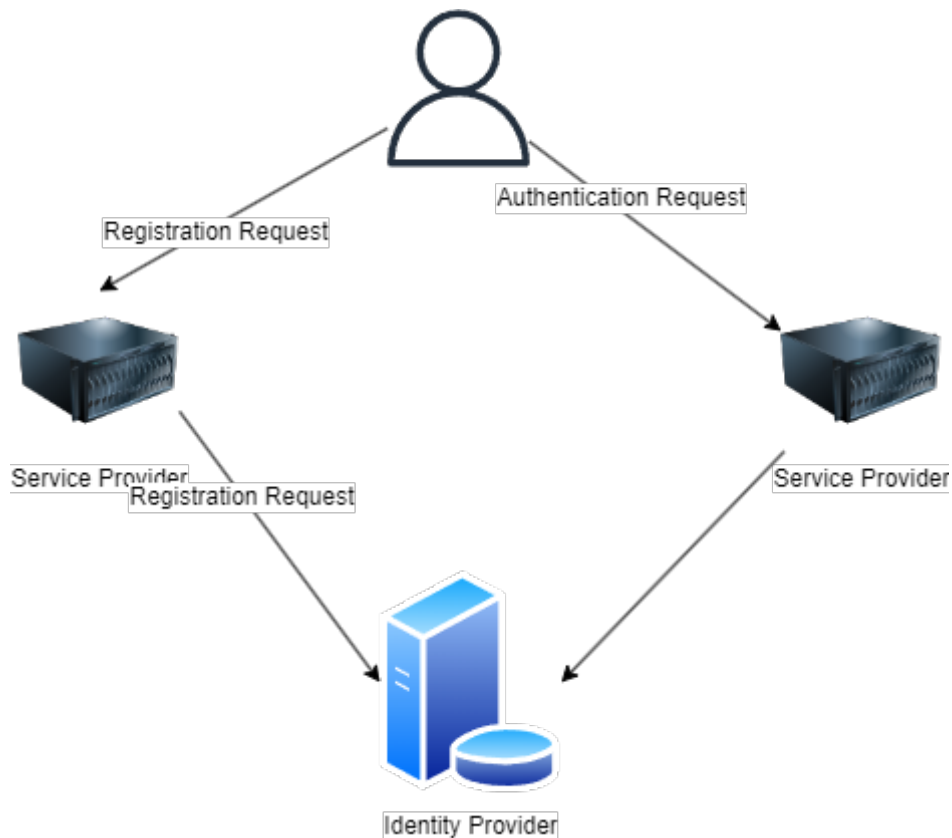


Figure 4: Centralized Model

2.2.1.1 Centralized model technologies

There are abundant frameworks for building a centralized identity system, many of them are built-in module in web framework. For example in Django ¹ consist of built-in models for Users and Groups which allow assignment of permission using flags.

2.2.1.2 Advantages of Centralized Model

The advantages of using a centralized identity system depends on the technologies that were used for the implementation. Generally speaking, implementing a centralized identity systems consume less time and offer somewhat more flexibility compared to other to other model of identity system.

2.2.1.3 Disadvantages of Centralized Model

According to (Smith and Khovratovich 2016), centralize models are vulnerable due to the fact that a single malicious could compromise the entire system of which all users data and credential can be stolen. Further the paper stated that users have to trust the good faith of the system administrators as they got privilege on any operations on users credentials including

¹<https://developer.mozilla.org/en-US/docs/Learn/Server-side/Django/Authentication>

modification or deletion. Finally the centralized of storage of user credential mean that it is more susceptible to hackers attack as it is often easier to attack a single point of location than multiples one.

2.2.2 Federated Model

The Federated model utilize a cross domain protocols for example oauth, OpenId that allow IdPs from different domains to recognize authentications from other Idps. It is important to note that users could still have different identifier in different domain. As figure shown, the federated model have the same components namely users, SPs and Ips. The main difference between federated model and centralized model is that Idps are sharing information with each other using standard protocol. (Cao and Yang 2010)

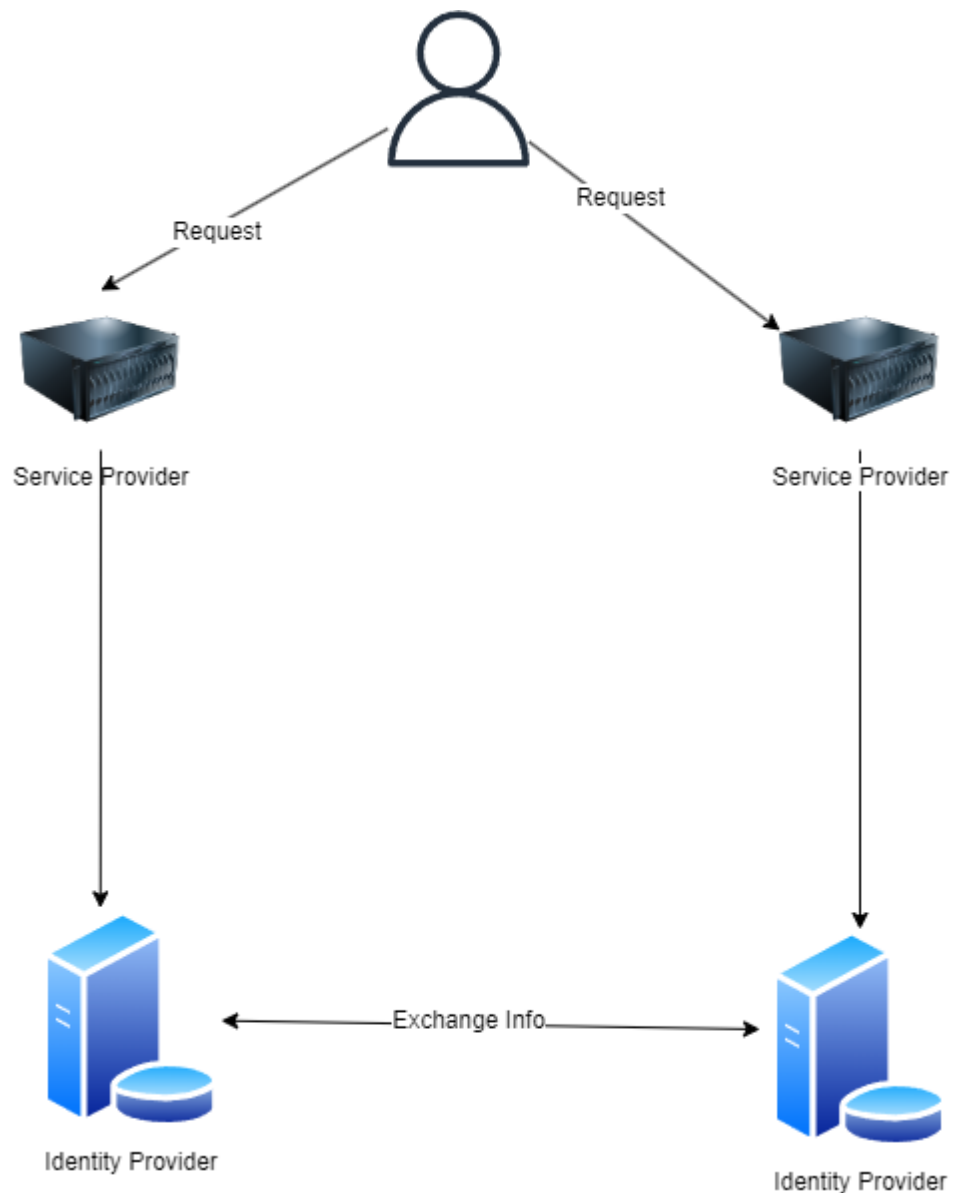


Figure 5: Federated Model

Figure 5 illustrates the components within the federated identity model. A significant distinction between the centralized and federated identity models is the presence of multiple identity providers within the federation. These providers facilitate the exchange of user information through standard protocols, enabling authentication interoperability across different domains.

2.2.2.1 Federated model technologies

The three major technologies for federated model are Security Assertion Markup Language (SAML), Open Authentication(OAuth), and OpenID Connect (OIDC). Majority of existing federated identity system are based on these three technologies. These technologies are not exclusive and often compliment each other, SAML is used in secure transportation user attributes across different domains within the federation. while Oauth is a authorization protocol that allow user to grant access to application, this is important as it allow automated workflow. Finally OIDC provide a standard interface for communicate Identity via RESTful API. OIDC builds upon OAuth 2.0 to provide authentication capabilities. In this study, Oauth server was used in the experiment therefore the primary focus will be on the Oauth protocol instead of the other two. Oauth is a scalable protocol that are used for delegation of authorization i.e. it allow users to grant permission for third party to act on their behalf via HTTP requests. The authorization was done via a token which were issue and distributed by users to the third party. The token can be in the following format: XML, JSON, JWT. Oauth2 is the latest version of Oauth and it consist of multiple specification, the most important one are the following:

- Core Spec
This spec describe the operation workflow of authorization.
- JWT Spec
JWT Spec describe the process of using JWT token to request Oauth token and for client authentication. JWT token is a secured token consist of identity information based on JSON.
- SAML Spec
SAML Spec describe the process of using SAML token to request Oauth token and for client authentication.

(Naik and Jenkins 2017)

2.2.2.2 Advantages of Federated Model

Federated identity systems allow secure identity resources sharing among partners within the federation in heterogeneous IT environments. For example when employee of company A are

transfer to subsidiary of company A for a temporary assignment, IdM team at the subsidiary do not need to recreate account for such employee again. For businesses, especially multi-national corporation this remove the duplicate identity management which in turn reduce the maintenance costs. Additionally it reduce the risk of human errors during identity transfer as the system now handle it automatically without the need of information being manually enter again. (Jensen 2012)

2.2.2.3 Disadvantages of Federated Model

For federated model, scalability issue had been encounter in scenario where there are many federation and servers. For example, (Pöhn and Hommel 2020) show that the growing number of IdPs and SPs is a well-known problem of SAML in academic world. Another common issue is the need of more fine-grained control on user role.

2.2.3 Decentralized Model

The decentralized model often called as self-sovereign model, has major differences from the centralized/federated model based on the fact that user control their own credential in their digital wallet instead of stored on remote databases that is controlled and accessible by other entity. This is made possible due to the use of public/private key cryptography and the blockchain technologies. The decentralized model consist of the same conceptual components but as figure shown the IdP has delegated some tasks to users and SPs such as the storage of credentials. (Preukschat et al. 2021)

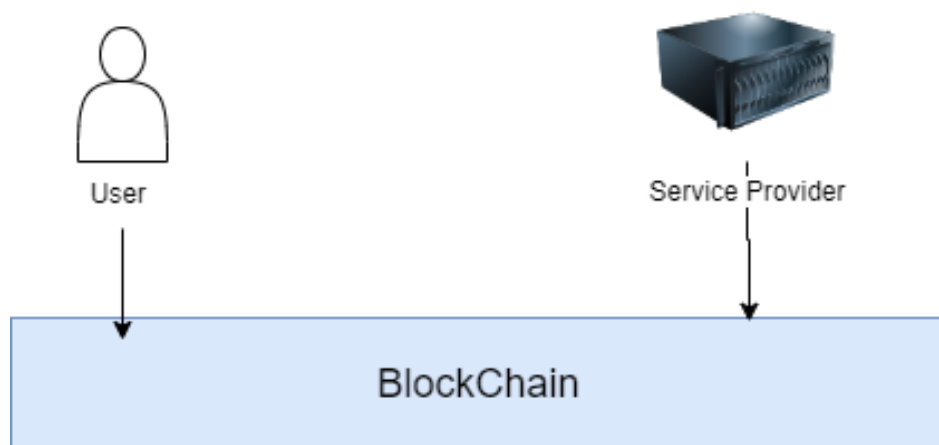


Figure 6: Decentralized Model

Figure 6 depicts the decentralized identity model, characterized by its distributed Identity Providers (IdPs). In this model, the verification of credentials is performed by the blockchain using public keys, while private keys are stored with users and service providers. As a result, users and service providers assume equal roles within the system, and the IdPs operate in

a trustless manner, ensuring that no single entity can modify stored information due to the immutable nature of the blockchain. This resistance to data tampering enhances the security and reliability of the decentralized identity model.

2.2.3.1 Decentralized model technologies

The essential building block that enable decentralized is the distributed ledger. Blockchain has been one of the most widely use technology for implementation of distributed ledger due to its properties of decentralization, immutability and transparency. Some of the existing systems include the following:

- uPort² - an open source permissionless Identity solution based on Ethereum
- Sovrin³ - an open source permissioned Identity solution based on Hyperledger Indy

Decentralized Identifiers (DIDs) and Verifiable Credentials (VCs) are two essential concepts for blockchain based decentralized solution. Decentralized Identifiers used for credential exchange and authentication and has a private key association with each DIDs. Verifiable Credentials define the format for credential and provide a cryptographically signed statement about a subject by its issuer.

The private key and credentials are managed by application called wallets which are often in the form of mobile apps or web browser extension. (Dib and Toumi 2020)

2.2.3.2 Advantages of Decentralized Model

As identity data are stored with users and managed by users, it eliminate the need of trusted intermediary. This made it more difficult for malicious actors to attempt hacking as the system do not have a single point of storage of user credential as seen in other identity model. The privacy of users are also protected as information that are not necessary are not shown to other party. (Kubach et al. 2020)

2.2.3.3 Disadvantages of Decentralized Model

Users become the sole responsibility party of their credentials. If credentials are lost, there would be no account recovery. Thus it is critical that users keep their credentials safe. Secondly the standard for decentralized identity model are constantly evolving as it is a relatively new technology. Thirdly utilizing decentralized identity model would mean the demonetization of user data for business, it remain to be seen that if alternative business model can be arise for decentralized identity systems. (ibid.)

²<https://github.com/uport-project>

³<https://github.com/sovrin-foundation/sovrin>

2.3 Software defined power measurement

Developers often shy away from gaining deeper insights into the power consumption of the software they code due to the hardware requirements for power consumption measurement. To address this, a software-based approach to power consumption measurement has been developed. There are two main approaches to software-defined power meters. The first approach uses the built-in Linux power capping, which requires processors with this capability. The second approach uses empirical polynomial regression models that were created using benchmarked data from hardware power meter measurement and CPU utilization calculated from CPU cycles obtained from the /proc/stat system file. The reported error rate of using PowerJoular’s regression models is around 3%. PowerJoular can also monitor the power consumption of individual applications by selecting their process ID. (Noureddine 2022)

2.4 Comparison between Identity management system models

Table 2: Comparison of Identity Management Models

Model	Centralized	Decentralized	Federated
Characteristics	Single authority for authentication and identity storage	Multiple authorities for authentication and identity storage	Multiple authorities for authentication and identity storage
User Control	Low	High	Medium
Data Ownership	Central authority	Individual users	Distributed among multiple authorities
Scalability	Limited scalability	Scalable	Scalable
Privacy	Privacy concerns due to centralization	User-controlled sharing of data	User-controlled sharing of data
Integration	Centralized integration	Diverse integration	Federated integration
Measurability	Easy	Hard	Hard
Example	Active Directory	Blockchain-based identity systems	OAuth/OpenID Connect

Table 2 offers a concise overview of the key characteristics and distinctions among Centralized, Decentralized, and Federated identity management models.

In terms of authentication and identity storage, the Centralized model relies on a single authority, while the Decentralized and Federated models involve multiple authorities.

Regarding user control, the Decentralized model offers the highest level of control, followed by the Federated model, while the Centralized model has the lowest level.

Data ownership differs across the models. In the Centralized model, the central authority owns the data, whereas in the Decentralized model, individual users have ownership. The Federated model involves the distribution of data among multiple authorities.

Scalability is considered scalable for both the Decentralized and Federated models, while the Centralized model has limited scalability. Privacy concerns arise due to centralization in the Centralized model, whereas the Decentralized and Federated models allow for user-controlled sharing of data. Integration varies among the models. The Centralized model involves centralized integration, the Decentralized model requires diverse integration, and the Federated model employs federated integration.

In terms of measurability, the centralized model is expected to be the most straightforward to measure power consumption. This is because all the relevant components are located in one centralized location, enabling easy and direct power monitoring. On the contrary, the decentralized and federated models introduce distributed architectures and dynamic interactions, making the measurement of power consumption more challenging. Monitoring power usage in these models involves the complexities of observing multiple nodes and intricate communication patterns, which can introduce additional hurdles to accurate measurements. Lastly, examples of each model are provided: Active Directory represents the Centralized model, blockchain-based identity systems represent the Decentralized model, and OAuth/OpenID Connect exemplify the Federated model.

In chapter 3, implementation of systems based on these three models are provided.

3 METHODOLOGY

In this chapter, a detail explanation of how the study are conducted are provided.

3.1 Methodology Overview

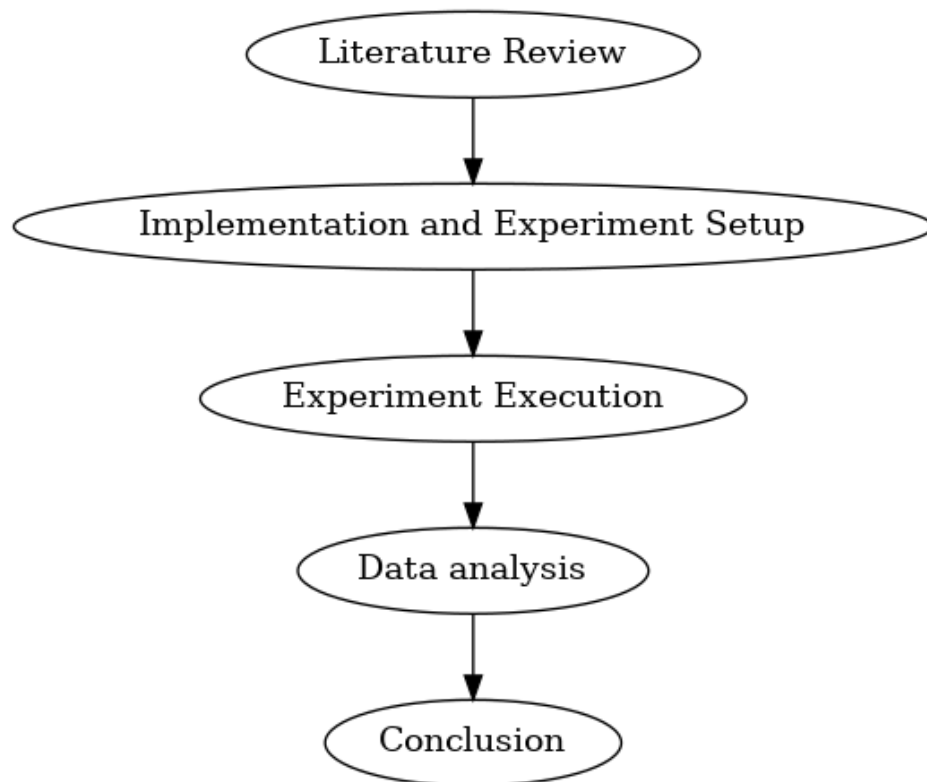


Figure 7: Flow Chart for Methodology

In this study, a literature review was conducted to gather a wide range of information on relevant technologies. The focus then shifted to the implementation and experiment setup, where identity solutions were implemented using three different IdM models within dockers. To measure power consumption, experiments were executed using a software-defined power measurement tool, capturing valuable data. The collected data was analyzed using statistical tests to derive meaningful insights. Finally, based on the analysis and findings, conclusive answers were provided to address the research questions, wrapping up the study’s conclusions. Figure 7 give a summary overview of the methodology used in this study.

Table 3: Research Variables

	Rate of operations	Type of IdM	Power in Watt
Independent Variable	X	X	
Dependent Variable			X

3.2 Experimental Design

3.2.1 Research Variables

Research variable are divided into independent and dependent variables. Independent variables represent variables that can be changed or manipulated in the experiment while dependent variable represent the outcome of the experiment. Table 3 stated the research variables of this study.

3.2.2 Experiment Environment

All software artefacts of the subjects are executed and run in docker containers, there are several reason for this. First the ease of deployment with containers mean that experimental replication are simple and efficient, secondly the use of docker containers allow power consumption measurement for individual components within the subject. All experiments are executed on a Laptop with AMD Ryzen 7 5800H 3.20GHz processor and 16 GB RAM.

3.2.2.1 Independent assumption of data

In addition of running multiple times of the experiment, all components of the subject are deployed in docker containers. This allow a clean environment for deployment and remove any residue impact of previous test run and therefore ensure independence of the data collected.

3.2.3 Experimental Run Design

The experiment consist of experiment run of the following operations

- Registration
The process of establish identity of users in Idps and/or SPs
- Authentication
The process of verify the identity of user

As shown in Algorithm 1, there are three parameters input for the experiment, and explanation and actual values for these parameters are shown in table 4

Algorithm 1 Algorithm of the experimental run

```
procedure TestRun(operations, noTest, noOps)  
  for all t in noTest do  
    for all op in operations do  
      Collecting Power Consumption data  
      for all n in noOps do  
        Execute op  
      end for  
      End data collection  
    end for  
  end for  
end procedure
```

Table 4: Parameters of the experiment run

Parameters Name	operations	noTest	noOps	sendRate
Explanation	Type of operation that were executed during the test	Number of test run	Number of executed operation during a test run	Number of Request per second
Actual value used	Register, Login	10	1000	5, 10, 20, 50

3.2.4 Hypothesis Formulation

Table 5 provides hypotheses for three research questions related to power consumption in identity solutions. It presents both the null hypotheses (H0) and alternative hypotheses (Ha) for each research question. These hypotheses serve as the foundation for investigating the relationships between power consumption, identity solutions, and other factors.

Research Question	Null Hypothesis (H0)	Alternative Hypothesis (Ha)
RQ1	There are no significant differences in power consumption between the identity solutions.	There are significant differences in power consumption between the identity solutions.
RQ2	There is no relationship between send rate and power consumption in the identity solutions.	There is a positive relationship between send rate and power consumption in the identity solutions.
RQ3	There are no significant differences in power consumption for components within the identity solutions.	There are significant differences in power consumption for components within the identity solutions.

Table 5: Hypotheses for Research Questions

3.3 Identity system setup

In this section, it illustrate how the three type of identity solutions are setup and what has been done for a fair comparison.

The experiment consists of three subjects: centralised identity system, federated identity system, and decentralized identity system. The centralised identity system was built using the Python Flask framework. It consists of a server and a relational database each running in its container. Using the flask-login to implement a classic centralised identity system architecture.

3.3.1 Centralized identity system

The centralized identity system are built using Flask framework¹. Flask is a micro web framework, which mean that it focus solely on the application level. It is chosen due to its ease of implementation. The implemented system consist of two components, the flask backend server and postgre database. For the implementation of backend server, flask_login² library are used. This python library allow quick implementation of IdM operations and session management. For the implementation of database, postgre are chosen. The backend server consist of two http endpoints that were relevant to the study:

- Register

The register endpoint is implemented as a function which receive JSON HTTP POST request that consist of a email and a password. Then the email are checked against the postgre database to made sure there are no duplication since emails are served as ID

¹<https://flask.palletsprojects.com/en/2.3.x/>

²<https://flask-login.readthedocs.io/en/latest/>

in this particular IdM. After successful verification of no email duplication, the email and password are stored and a successful response are returned back to the client.

- Login

The Login endpoint is implemented as a function which receive JSON HTTP POST request that consist of a email and a password. The backend server first check if the email exist in the database, if not it returned a failure response to the client. Otherwise the server check the corresponding password and if it matches, a successful response is returned else a failure response.

The execution of operations are via RESTFUL API request.

3.3.2 Federated identity system

The Federated identity system chosen for the study are already implemented, distributed under The GNU General Public License and its source code³ are hosted on Github. This implementation are lightweight and implemented according to the OAuth2 specification. The two endpoints that were tested are register and login.

The Federated identity system consist of three components:

- Backend server - built using Echo⁴ a minimalist Go web framework
- Message Broker - built using Redis
- Database - MySQL

The backend server consist of two http endpoints that were relevant to the study:

- Register

The register endpoint is implemented as a function which receive JSON HTTP POST request that consist of a email and a password.

- Login

The Login endpoint is implemented as a function which receive JSON HTTP POST request that consist of a email and a password.

The execution of operations are via RESTFUL API request.

³<https://github.com/menduong/oauth2-server>

⁴<https://github.com/labstack/echo>

3.3.3 Decentralized identity system

For the decentralized identity system, Findy Agency⁵ was used. It is an open-source Hyperledger Aries compatible identity agent service project. Hyperledger Aries⁶ is a framework that help building decentralized identity blockchain infrastructure. Findy Agency provides consist of different components working together. The components relevant to this study are the following:

- findy-agent
This component handles all agent functionality including credential handling and Aries protocols
- findy-agent-auth
This component registers and authenticates all agency users

This two components are the main building block of the backend server. For client side, there are three tools that can be used and these are findy-agent-cli, findy-wallet-pwa and findy-agent-api. Findy-agent-cli offer client endpoint through the linux commandline interface while findy-agent-api offer through HTTP protocol. Findy-wallet-pwa is a demo webapp that allow users to interact through web user interface. In the experiment, the Findy-agent-cli are used to generate register and login requests to the backend.

The execution of register and login operation are done via Findy-agent-cli with the following commands respectively:

- FCLI authn register
- FCLI authn login

⁵<https://github.com/findy-network/findy-agent>

⁶<https://www.hyperledger.org/use/aries>

3.4 Summary of Identity system

Table 6: Implementation Details of Identity Solutions

System Type	Implementation Details
Centralized	Flask framework is used for implementation. Flask backend server with Flask-Login library for IdM operations and session management. PostgreSQL database is used.
Federated	Lightweight system implemented according to OAuth2 specification. GNU General Public License. Source code hosted on Github. Endpoints for register and login.
Decentralized	Findy Agency, an open-source Hyperledger Aries compatible identity agent service project. Built on Hyperledger Aries framework.

Table 7: Components of Identity Solutions

System Type	Backend	Database	Message Broker
Centralized	Flask	PostgreSQL	-
Federated	Golang	MySQL	Redis
Decentralized	findy-agent	findy-agent	-

Table 6 and 7 presents a comprehensive overview of different identity solutions, their implementation details, and the key components involved in each system.

The tables provides a summary of different identity solutions along with their implementation details and components. The first system type is "Centralized," which utilizes the Flask framework for implementation. It employs a Flask backend server with the Flask-Login library for IdM operations and session management. The system relies on a PostgreSQL database for storing data. The components of this system include a Flask backend server and a PostgreSQL database.

The second system type is "Federated." It is a lightweight system implemented based on the OAuth2 specification. The implementation is licensed under the GNU General Public License and hosted on Github. The system provides endpoints for user registration and login. It involves a Golang backend server, a MySQL database for data storage, and a Redis Message Broker for communication between components.

The third system type is "Decentralized." It employs the Findy Agency, an open-source project that serves as an identity agent service. It is built on the Hyperledger Aries framework, which provides compatibility for decentralized IdM. The components of this system include the findy-agent for the backend and database, and the findy-agent-auth for backend operations.

4 RESULT

In this chapter, the experimental result of power consumption of each type of Idm operations (register and login) are presented with comparison of each identity model type and comparisons of components within a single identity systems.

4.1 Collected Data

The power consumption data collection process involves a comprehensive testing approach to ensure reliable and statistically robust results. For each of the Identity Management (IdM) solutions, namely centralized, decentralized, and federated, a total of 10 test runs are conducted at each sendrate (5,10,20,50). In each test run, a set of 1000 operations, encompassing both registration and login activities, is executed. These operations represent typical usage scenarios to simulate real-world conditions accurately. The purpose of conducting multiple test runs is to capture variations in power consumption that may arise due to external factors or system fluctuations. To further enhance the credibility of the collected data, the tests are performed at four different send rates: 5, 10, 20, and 50 operations per second. These diverse send rates help evaluate the impact of varying workloads on the power consumption profiles of the IdM systems. It enables us to observe how the systems behave under low, moderate, and high load scenarios, providing a more comprehensive understanding of their power consumption characteristics. After each test run, the power consumption data from all components within the IdM systems are meticulously recorded. This data includes energy consumption measurements for servers, databases and any other relevant components involved in the IdM processes. To ensure the statistical robustness of the results, the average power consumption is calculated for each test scenario. For the rest of this section, data are presented in chart and graph in a descriptive manners.

4.1.1 Power Consumption for Register

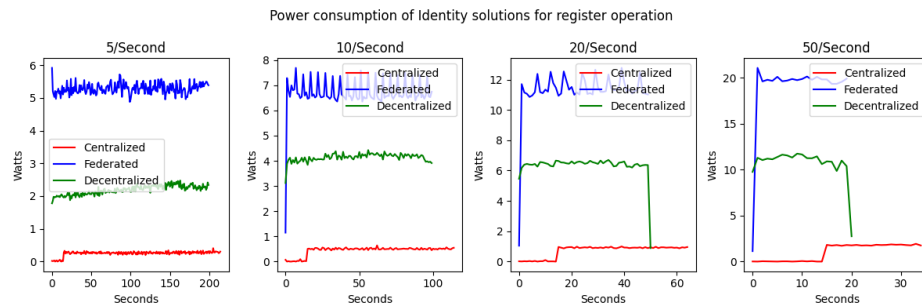


Figure 8: Power Consumption for Register Operations

Figure 8 shown the result for sending 5 register request per second and from the figure it can be shown that federated Identity consume most power at around 0.005 watt over period of 200 seconds in another words it uses 1 joules of energy for 1000 registration. On the contrary, decentralized consume 0.4 joule and centralized consume 0.06 Joule. Based on this data, it is can be shown that for registration federated IdM consume most power, followed by decentralized IdM and centralized IdM.

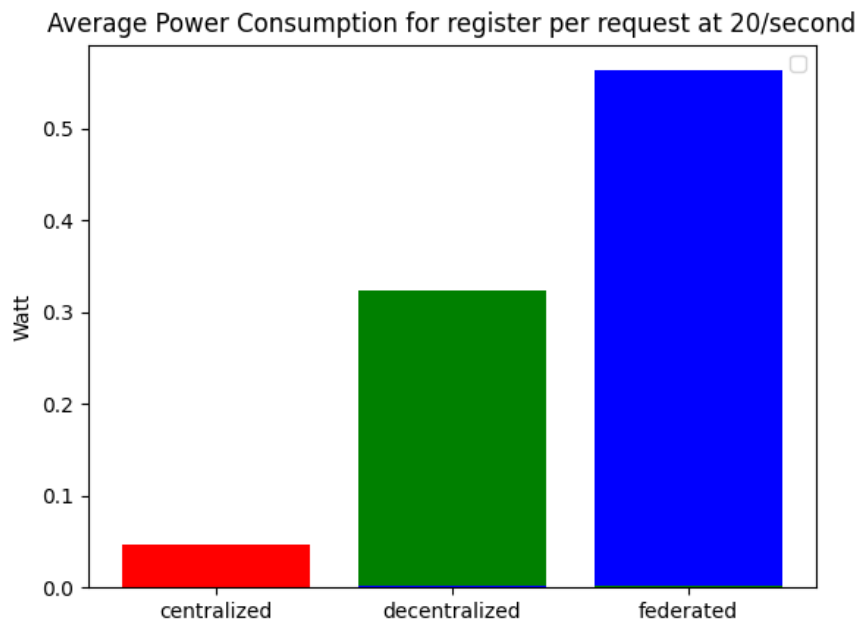


Figure 9: Power Consumption for Register 20/second Operation

Figure 9 shown the average power consumption for register request using the sendrate of 20 data set. It shown that centralized model consume the least power at around 0.05 watt, followed by decentralized model which consume about 0.3 watt and with federated model consume the most at around 0.55 watt per register request.

4.1.2 Power Consumption for Login

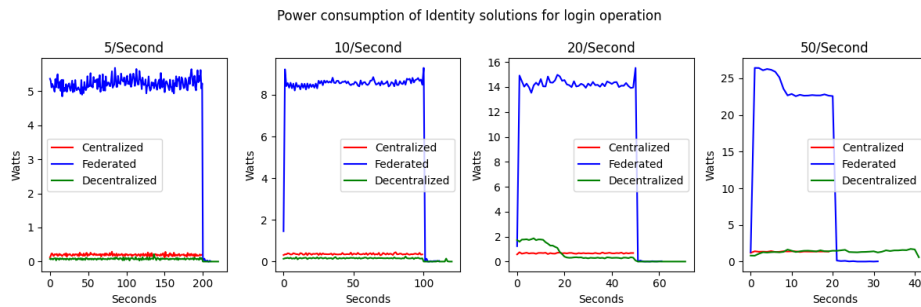


Figure 10: Power Consumption for Login Operation

Figure 10 shown the power consumption of identity systems for login operation at 5, 10, 20 and 50 request/second. Across all send rate, it can be see that federated consume most power, followed by centralized and decentralized model. It is interesting to noted that as the send rate get higher the time, the more power are consumed.

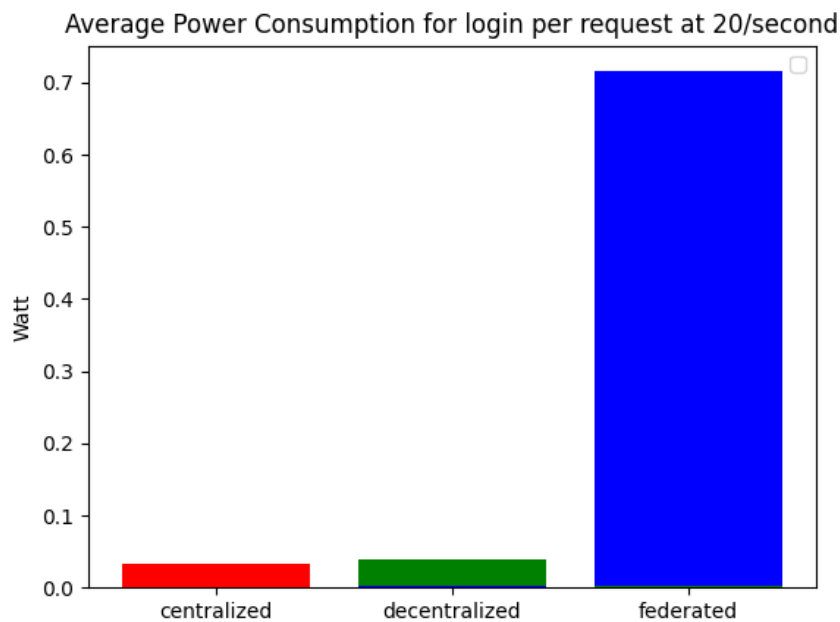


Figure 11: Power Consumption for Login 20/second Operation

Figure 11 shown the average power consumption for login operation at the send rate of 20/second. It can be see that centralized and decentralized model consume roughly the same power while federated model consume power more than 7 times than other model at 0.7 watt.

4.1.3 Power consumption for processes

4.1.3.1 Centralized

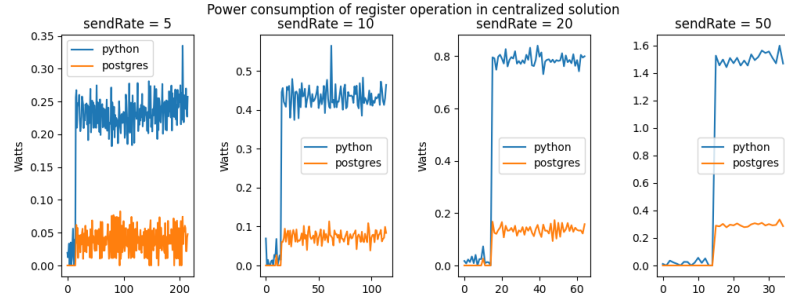


Figure 12: Power Consumption for Register operation in processes

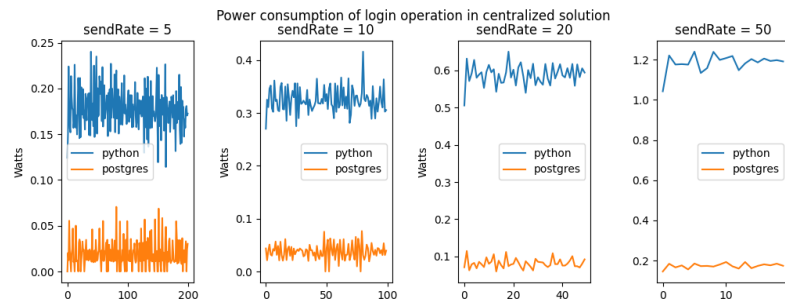


Figure 13: Power Consumption for Login operation in processes

Figures 12 and 13 display the power consumption data for processes within the centralized identity system at varying send rates (5, 10, 20, 50). Notably, Figure 12 reveals that the backend Python process, responsible for handling registration requests, consistently exhibits the highest power consumption across all send rates. The initial significant increase in power consumption is attributed to the server handling incoming requests during its startup phase.

Furthermore, Figure 12 illustrates that the backend server continues to consume more power than the database during the registration operation. This observation can be ascribed to the inherent nature of write operations, which typically require more time than read operations when interacting with the database. Consequently, it is evident that the registration operations take longer to execute compared to login operations.

It is noteworthy that as the send rate increases, the power consumption demonstrates a steady upward trend across all operations. This aligns with the expected behavior, as increased data processing and network activity lead to higher power requirements.

4.1.3.2 Decentralized

Figures 14 and 15 present the power consumption data for processes within the decentralized identity system at various send rates (5, 10, 20, 50). In this decentralized IdM model, a signif-

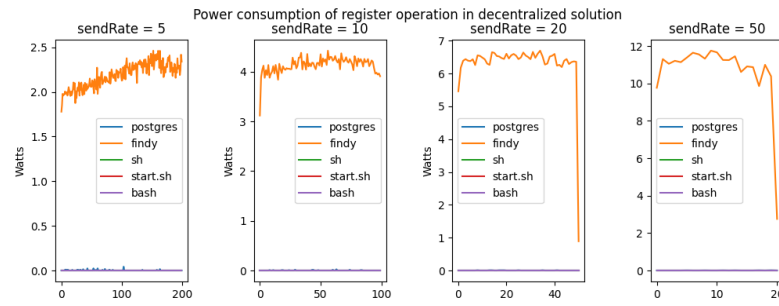


Figure 14: Power Consumption for Register operation in processes

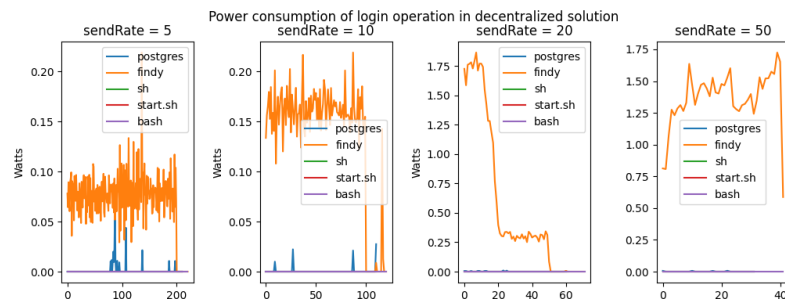


Figure 15: Power Consumption for Login operation in processes

icant process is "findy," which functions as the backend and database component. Notably, the users' data are stored with the respective users in this decentralized approach.

Across all send rates and operations, the "findy" process consistently exhibits the highest power consumption, while other processes consume negligible amounts of power.

Moreover, a notable observation is that the power consumption during the register operation is notably higher than during the login operation. For instance, at a send rate of 20, the register operation consumes approximately 6.5 watts over the test duration, whereas the login operations range from 1.75 watts to 0.25 watts.

The reduction in power consumption towards the end of the measurements may be attributed to the cleanup of tasks in the backlog, resulting in a lag between task completion and the end of power consumption measurements, ultimately leading to a decline in power consumption.

4.1.3.3 Federated

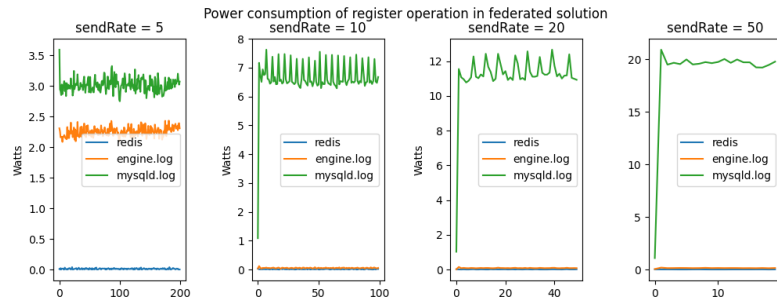


Figure 16: Power Consumption for Register operation in processes

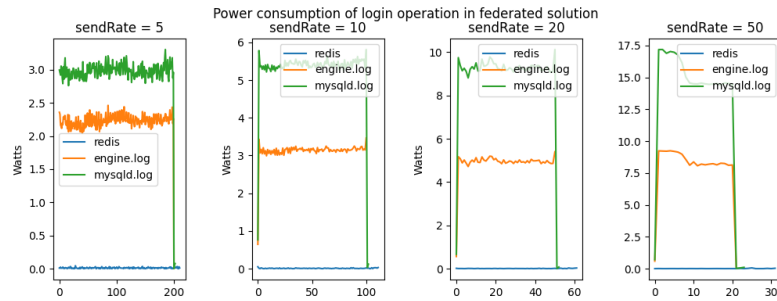


Figure 17: Power Consumption for Login operation in processes

Figures 16 and 17 present the power consumption data for various processes within the federated identity system, measured at different send rates (5, 10, 20, 50). Notably, the backend component, denoted as "engine.log," consistently exhibits the highest power consumption across all send rates and operations. Following closely are the "redis" component (message broker) and "mysqld.log" (database) with relatively lower power consumption.

The prominence of the message broker's power consumption can be attributed to its role in handling requests between the database and the backend, thereby reducing the database's power consumption while increasing its own. Additionally, the comparison between the power consumption of register and login operations reveals that the former consumes more power, though the difference is not as significant as observed in other IdM systems.

The observed sharp increase and decrease in power consumption at the beginning and end of the measurements can be attributed to the lag time between initiating power measurement and commencing requests, as well as concluding power measurement and completing request execution.

4.2 Statistical Test Analysis

4.2.1 Statistical test for RQ1

In this section, statistical tests on the raw data collected from the experiment to test the formulated hypothesis for research question 1.

H₀: There are no significant difference in power consumption between the identity systems.

H_a: There are significant difference in power consumption between the identity systems.

In conclusion, the null hypothesis is rejected based on the p-values obtained from statistical analysis. This signifies that there exist significant differences in power consumption among the different identity systems. Utilizing Tukey's HSD test, the findings reveal that federated identity systems exhibit the highest power consumption, followed by decentralized and centralized identity systems. The subsequent sections explain the statistical test result in more details.

4.2.1.1 ANOVA statistical test analysis

Send Rate	F statistic	P-Value	Numer. d.f.	Denom. d.f.	F critical value
5	84121.28	0	2	582	3.0112
10	16704.23	1.278332e-303	2	282	3.0278
20	9756.12	5.42658e-155	2	133	3.0642
50	2889.86	1.476912e-56	2	43	3.2145

Table 8: ANOVA test for Register Operation between identity systems

Send Rate	F statistic	P-Value	Numer. d.f.	Denom. d.f.	F critical value
5	6027.93	0	2	599	3.0108
10	1502.69	4.842690e-167	2	300	3.0258
20	342.81	7.27454e-67	2	151	3.056
50	44.84	1.052736e-22	2	61	3.1478

Table 9: ANOVA test for Login Operation between identity systems

As Table 8 shown for the Register Operation: The F statistic values for different send rates (5, 10, 20, and 50) are 84121.28, 16704.23, 9756.12, and 2889.86, respectively. The corresponding p-values are 0, 1.278332e-303, 5.42658e-155, and 1.476912e-56, respectively. The numerator degrees of freedom (Numer. d.f.) for all cases are 2, representing the number of groups being compared. The denominator degrees of freedom (Denom. d.f.) are 582, 282,

133, and 43 for the respective send rates. The F critical values, representing the critical value for significance at a given alpha level, are 3.0112, 3.0278, 3.0642, and 3.2145.

For the Login Operation Shown by the table 9: The F statistic values for different send rates (5, 10, 20, and 50) are 6027.93, 1502.69, 342.81, and 44.84, respectively. The corresponding p-values are 0, 4.842690e-167, 7.27454e-67, and 1.052736e-22, respectively. The numerator degrees of freedom (Numer. d.f.) for all cases are 2, representing the number of groups being compared. The denominator degrees of freedom (Denom. d.f.) are 599, 300, 151, and 61 for the respective send rates. The F critical values, representing the critical value for significance at a given alpha level, are 3.0108, 3.0258, 3.056, and 3.1478.

Table 8 and 9 shown the ANOVA test result for the experiment result. All P-values are below the chosen significant level with alpha value at 0.05 and the corresponding confidence level is 95% which mean that across all categories there are significant difference in power consumption between the three group. Therefore the null hypothesis for RQ1 is rejected. The comparison between F-statistic value and the F-critical value provide more information about the magitude of the difference in power consumption between the groups. Across all categories f-statistic values are higher than f critical value more than tenfold and therefore it is a indicator that the difference are quite large and this also provide additional result to reject the RQ1 hypothesis.

4.2.1.2 Tukey's HSD Pairwise test analysis

Comparison	Statistic	p-value	Lower CI	Upper CI
(centralized - decentralized)	-1.933	0.000	-1.962	-1.904
(centralized - federated)	-5.023	0.000	-5.052	-4.994
(decentralized - centralized)	1.933	0.000	1.904	1.962
(decentralized - federated)	-3.090	0.000	-3.119	-3.060
(federated - centralized)	5.023	0.000	4.994	5.052
(federated - decentralized)	3.090	0.000	3.060	3.119

Table 10: Tukey's HSD Pairwise Group Comparisons for 5 register/sec

Comparison	Statistic	p-value	Lower CI	Upper CI
(centralized - decentralized)	-3.702	0.000	-3.784	-3.621
(centralized - federated)	-6.269	0.000	-6.351	-6.188
(decentralized - centralized)	3.702	0.000	3.621	3.784
(decentralized - federated)	-2.567	0.000	-2.652	-2.482
(federated - centralized)	6.269	0.000	6.188	6.351
(federated - decentralized)	2.567	0.000	2.482	2.652

Table 11: Tukey's HSD Pairwise Group Comparisons for 10 register/sec

Comparison	Statistic	p-value	Lower CI	Upper CI
(centralized - decentralized)	-5.717	0.000	-5.900	-5.535
(centralized - federated)	-10.744	0.000	-10.928	-10.560
(decentralized - centralized)	5.717	0.000	5.535	5.900
(decentralized - federated)	-5.027	0.000	-5.223	-4.830
(federated - centralized)	10.744	0.000	10.560	10.928
(federated - decentralized)	5.027	0.000	4.830	5.223

Table 12: Tukey's HSD Pairwise Group Comparisons for 20 register/sec

Table 10 presents the analysis for 5 register/sec, where it can be observed that the centralized system has a significantly lower mean compared to both the decentralized and federated systems, with statistics of -1.933 and -5.023, respectively. Similarly, in Table 11, for 10 register/sec, the centralized system exhibits significantly lower means compared to the decentralized and federated systems, with statistics of -3.702 and -6.269, respectively.

Moving to Table 12, which represents the analysis for 20 register/sec, it becomes evident that the centralized system continues to have significantly lower means compared to the decentralized and federated systems, with statistics of -5.717 and -10.744, respectively. Finally, Table 13 shows the results for 50 register/sec, where the centralized system once again demonstrates significantly lower means compared to both the decentralized and federated systems, with statistics of -10.265 and -18.807, respectively.

Comparison	Statistic	p-value	Lower CI	Upper CI
Centralized - Decentralized	-10.265	0.000	-10.869	-9.662
Centralized - Federated	-18.807	0.000	-19.431	-18.183
Decentralized - Centralized	10.265	0.000	9.662	10.869
Decentralized - Federated	-8.542	0.000	-9.270	-7.813
Federated - Centralized	18.807	0.000	18.183	19.431
Federated - Decentralized	8.542	0.000	7.813	9.270

Table 13: Tukey's HSD Pairwise Group Comparisons for 50 register/sec

In summary, across different register rates, the Tukey's HSD pairwise test results consistently highlight that the centralized system exhibits significantly lower means compared to both the decentralized and federated systems. These observed differences are statistically significant, with p-values of 0.000, providing a high level of confidence in the distinct performance of the systems.

4.2.2 Statistical test for RQ2

In this section, statistical tests on the raw data collected from the experiment to test the formulated hypothesis for research question 2.

H0: There are no relationship between send rate and power consumption in the identity systems.

Ha: There are positive relationship between send rate and power consumption in the identity systems.

To summarized, the null hypothesis rejected therefore there are positive correlation between send rate and power consumption in all IdM systems. The rest of the sections explain the statistical test result in more details.

4.2.2.1 ANOVA statistical test analysis

Identity system type	F statistic	P-Value	Numer. d.f.	Denom. d.f.	F critical value
Centralized	102.49	8.17552e-59	3	386	2.628
Decentralized	24699.83	0	3	328	2.6321
Federated	11304.42	0	3	326	2.6323

Table 14: ANOVA test for register operation at different send rate (5,10,20,50)

Identity system type	F statistic	P-Value	Numer. d.f.	Denom. d.f.	F critical value
Centralized	7267.84	3.805441e-308	3	326	2.6323
Decentralized	346.1	3.51828e-122	3	412	2.6266
Federated	140.2	7.94570e-71	3	373	2.6288

Table 15: ANOVA test for login operation at different send rate (5,10,20,50)

The ANOVA test results presented in the tables provide insights into the relationship between send rate and power consumption in different identity systems. The identity system types are categorized as Centralized, Decentralized, and Federated. The F statistic measures the variability between the means of the different identity system types. Higher F values indicate a stronger relationship between the independent variable (send rate) and the dependent variable (power consumption).

Table 14 - ANOVA test for the register operation: For the register operation, all three identity systems (Centralized, Decentralized, and Federated) show significant relationships between send rate and power consumption. The F statistics are high, indicating substantial variability. The P-values are very close to zero, indicating strong evidence against the null hypothesis (H0) and in favor of the alternative hypothesis (Ha). The F critical values provide a threshold for the F statistic to determine statistical significance.

Table 15 - ANOVA test for the login operation: Similar to the previous table, this table analyzes the relationship between send rate and power consumption for the login operation in different identity systems. Once again, all three identity systems (Centralized, Decentralized, and Federated) exhibit significant relationships between send rate and power consumption. The F statistics are relatively high, indicating notable variability. The P-values are close to zero, indicating strong evidence against the null hypothesis (H_0) and in favor of the alternative hypothesis (H_a). The F critical values provide a threshold for the F statistic to determine statistical significance.

In summary, the ANOVA test results suggest that there are positive relationships between send rate and power consumption in the identity systems for both the register and login operations. The findings are statistically significant, indicating that changes in the send rate have a significant impact on power consumption in the studied identity systems.

4.2.2.2 Tukey's HSD Pairwise tests

4.2.2.2.1 Centralized

Send Rate Comparison	Statistic	p-value	Lower CI	Upper CI
(5 - 10)	-0.202	0.000	-0.286	-0.118
(5 - 20)	-0.501	0.000	-0.608	-0.395
(5 - 50)	-0.824	0.000	-0.972	-0.675
(10 - 5)	0.202	0.000	0.118	0.286
(10 - 20)	-0.299	0.000	-0.416	-0.182
(10 - 50)	-0.621	0.000	-0.777	-0.465
(20 - 5)	0.501	0.000	0.395	0.608
(20 - 10)	0.299	0.000	0.182	0.416
(20 - 50)	-0.322	0.000	-0.491	-0.153
(50 - 5)	0.824	0.000	0.675	0.972
(50 - 10)	0.621	0.000	0.465	0.777
(50 - 20)	0.322	0.000	0.153	0.491

Table 16: Tukey's HSD Pairwise Group Comparisons for Centralized system (Register)

Send Rate Comparison	Statistic	p-value	Lower CI	Upper CI
(5 - 10)	-0.164	0.000	-0.174	-0.155
(5 - 20)	-0.472	0.000	-0.486	-0.459
(5 - 50)	-1.171	0.000	-1.196	-1.147
(10 - 5)	0.164	0.000	0.155	0.174
(10 - 20)	-0.308	0.000	-0.322	-0.294
(10 - 50)	-1.007	0.000	-1.032	-0.982
(20 - 5)	0.472	0.000	0.459	0.486
(20 - 10)	0.308	0.000	0.294	0.322
(20 - 50)	-0.699	0.000	-0.726	-0.672
(50 - 5)	1.171	0.000	1.147	1.196
(50 - 10)	1.007	0.000	0.982	1.032
(50 - 20)	0.699	0.000	0.672	0.726

Table 17: Tukey's HSD Pairwise Group Comparisons for Centralized system (Login)

The Tukey's HSD pairwise group comparisons were performed to analyze the significant differences between different send rates in the Centralized system for both the Register and Login operations.

In the Register operation (Table 16), the statistical analysis shows that there are significant differences between the groups across all comparisons. The p-values are all reported as 0, indicating a highly significant result. The lower and upper confidence intervals (CIs) provide a range of the mean differences between the groups, and they do not overlap, further supporting the significant differences.

Similarly, in the Login operation (Table 17), the statistical analysis reveals significant differences between the groups for all comparisons. The p-values are reported as 0, indicating a high level of significance. The CIs also do not overlap, further confirming the significant differences between the groups.

Overall, these findings indicate that there are statistically significant differences in performance across different send rates in the Centralized system for both Register and Login operations.

4.2.2.2.2 Decentralized

Send Rate Comparison	Statistic	p-value	Lower CI	Upper CI
(5 - 10)	-1.972	0.000	-2.017	-1.926
(5 - 20)	-4.286	0.000	-4.347	-4.224
(5 - 50)	-9.156	0.000	-9.266	-9.046
(10 - 5)	1.972	0.000	1.926	2.017
(10 - 20)	-2.314	0.000	-2.381	-2.247
(10 - 50)	-7.184	0.000	-7.298	-7.071
(20 - 5)	4.286	0.000	4.224	4.347
(20 - 10)	2.314	0.000	2.247	2.381
(20 - 50)	-4.870	0.000	-4.991	-4.750
(50 - 5)	9.156	0.000	9.046	9.266
(50 - 10)	7.184	0.000	7.071	7.298
(50 - 20)	4.870	0.000	4.750	4.991

Table 18: Tukey's HSD Pairwise Group Comparisons for Decentralized system (Register)

Send Rate Comparison	Statistic	p-value	Lower CI	Upper CI
(5 - 10)	-0.066	0.070	-0.135	0.004
(5 - 20)	-0.424	0.000	-0.510	-0.339
(5 - 50)	-1.334	0.000	-1.446	-1.222
(10 - 5)	0.066	0.070	-0.004	0.135
(10 - 20)	-0.359	0.000	-0.452	-0.265
(10 - 50)	-1.268	0.000	-1.387	-1.149
(20 - 5)	0.424	0.000	0.339	0.510
(20 - 10)	0.359	0.000	0.265	0.452
(20 - 50)	-0.909	0.000	-1.038	-0.781
(50 - 5)	1.334	0.000	1.222	1.446
(50 - 10)	1.268	0.000	1.149	1.387
(50 - 20)	0.909	0.000	0.781	1.038

Table 19: Tukey's HSD Pairwise Group Comparisons for Decentralized system (login)

The Tukey's HSD pairwise group comparisons for the decentralized system (register and login operations at different send rates) are presented in Tables 18 and 19, respectively.

In the decentralized system for the register operation, the Tukey’s HSD test reveals significant differences between the send rates. The comparison of send rates 5 and 50 shows the highest negative statistic (-9.156) and a p-value of 0.000, indicating a significant difference in means. Similarly, the comparisons between send rates 10 and 50, as well as 20 and 50, also exhibit significant differences with negative statistics (-7.184 and -4.870, respectively) and p-values of 0.000. On the other hand, the comparisons between send rates 5 and 10, as well as 10 and 20, demonstrate smaller negative statistics but still have p-values of 0.000, indicating significant differences in means.

For the decentralized system in the login operation, the Tukey’s HSD test results show significant differences among the send rates. The comparisons of send rates 5 and 50, 10 and 50, and 20 and 50 exhibit the highest negative statistics (-1.334, -1.268, and -0.909, respectively) with p-values of 0.000, indicating significant differences in means. Similarly, the comparisons between send rates 5 and 20, as well as 10 and 20, also demonstrate significant differences with negative statistics and p-values of 0.000. The comparisons involving send rate 10 show the smallest negative statistics, but still indicate significant differences in means with p-values of 0.000.

Overall, the Tukey’s HSD pairwise group comparisons suggest that there are significant differences in means across different send rates in both the register and login operations of the decentralized system.

4.2.2.2.3 Federated

Send Rate Comparison	Statistic	p-value	Lower CI	Upper CI
(5 - 10)	-1.448	0.000	-1.547	-1.350
(5 - 20)	-6.222	0.000	-6.357	-6.088
(5 - 50)	-14.607	0.000	-14.858	-14.357
(10 - 5)	1.448	0.000	1.350	1.547
(10 - 20)	-4.774	0.000	-4.920	-4.627
(10 - 50)	-13.159	0.000	-13.416	-12.901
(20 - 5)	6.222	0.000	6.088	6.357
(20 - 10)	4.774	0.000	4.627	4.920
(20 - 50)	-8.385	0.000	-8.658	-8.112
(50 - 5)	14.607	0.000	14.357	14.858
(50 - 10)	13.159	0.000	12.901	13.416
(50 - 20)	8.385	0.000	8.112	8.658

Table 20: Tukey’s HSD Pairwise Group Comparisons for Federated system (Register)

The Tukey’s HSD test was performed on the data obtained from the federated system for the login and register operations at different send rates. The test results are presented in two tables: one for the login operation and the other for the register operation.

Send Rate Comparison	Statistic	p-value	Lower CI	Upper CI
(5 - 10)	-2.995	0.000	-4.027	-1.962
(5 - 20)	-7.557	0.000	-8.878	-6.235
(5 - 50)	-11.902	0.000	-13.809	-9.994
(10 - 5)	2.995	0.000	1.962	4.027
(10 - 20)	-4.562	0.000	-6.009	-3.115
(10 - 50)	-8.907	0.000	-10.903	-6.911
(20 - 5)	7.557	0.000	6.235	8.878
(20 - 10)	4.562	0.000	3.115	6.009
(20 - 50)	-4.345	0.000	-6.505	-2.185
(50 - 5)	11.902	0.000	9.994	13.809
(50 - 10)	8.907	0.000	6.911	10.903
(50 - 20)	4.345	0.000	2.185	6.505

Table 21: Tukey's HSD Pairwise Group Comparisons for Federated system (Login)

For the login operation, the pairwise group comparisons indicate significant differences in power consumption among all send rate combinations (5, 10, 20, and 50). The p-values associated with all comparisons are reported as 0.000, indicating strong evidence against the null hypothesis of no significant difference. The statistic values for each comparison indicate the magnitude of the difference, with negative values representing lower power consumption and positive values representing higher power consumption.

In the register operation, similar findings are observed. All pairwise comparisons between send rates exhibit significant differences in power consumption, as indicated by the p-values of 0.000. The statistic values reveal the direction and magnitude of the differences, with negative values suggesting lower power consumption and positive values indicating higher power consumption.

Overall, the Tukey's HSD test demonstrates that there are statistically significant differences in power consumption among the different send rates for both login and register operations in the federated system. The results suggest that certain send rates may consume more or less power compared to others.

4.2.3 Statistical test for RQ3

In this section, statistical tests on the raw data collected from the experiment to test the formulated hypothesis for research question 3.

H0: There are no significant difference in power consumption for components within the identity systems.

Ha: There are significant difference in power consumption for components within the identity systems.

In summary, the null hypothesis is rejected, indicating a significant difference in components within all Identity Management (IdM) systems. Specifically, the backend server is found to be the most power-consuming component. Further elaboration on the remaining sections is provided in subsequent discussions.

4.2.3.1 ANOVA statistical test analysis

In this section, the ANOVA statistical test was conducted to analyze the power consumption differences among components within the identity systems. The formulated hypothesis for research question 3 was tested, where the null hypothesis (H0) stated that there are no significant differences in power consumption, and the alternative hypothesis (Ha) stated that there are significant differences.

4.2.3.1.1 Test result for centralized identity system

In the implemented centralized identity system, it consist of two processes Python and Postgres which corresponding to the backend server and the database server.

Send Rate	F statistic	P-Value	Numer. d.f.	Denom. d.f.	F critical value
5	1845.97	2.558492e-167	1	328	3.8633
10	537.41	6.908457e-72	1	228	3.8826
20	150.43	2.386150e-23	1	128	3.9151
50	29.46	8.23250e-17	1	68	3.9819

Table 22: ANOVA test for register processes of centralized Identity system

Send Rate	F statistic	P-Value	Numer. d.f.	Denom. d.f.	F critical value
5	6766.89	6.36884e-262	1	398	3.8449
10	10506.18	1.589006e-183	1	198	3.8889
20	14812.16	9.46374e-119	1	98	3.9381
50	10227.02	8.1284e-58	1	38	4.0982

Table 23: ANOVA test for login processes of centralized Identity system

The test results for the centralized identity system were examined. The system consisted of two processes, Python and Postgres, representing the backend server and the database server,

respectively. The ANOVA test was performed separately for the register and login processes, and the results are shown in Tables 22 and 23.

For the register processes, at a 0.05 significance level, the p-values were all extremely small (e.g., 2.558492e-167), indicating strong evidence against the null hypothesis. The F statistics were large (e.g., 1845.97), suggesting significant differences among the components. All F statistics value are greater than F critical values. As a example at send rate of 5 request per second, the degrees of freedom for the numerator and denominator were 1 and 328, respectively. The critical value for the F statistic at the chosen significance level was 3.8633. Similar patterns were observed for the login processes, with low p-values, high F statistics, and significant differences.

These findings provide strong evidence to reject the null hypothesis, indicating that there are indeed significant differences in power consumption among the components within the centralized identity system. The power consumption variations observed may have implications for optimizing and improving the performance of the identity system components.

4.2.3.1.2 Test result for decentralized identity system

In the implemented decentralized identity system, it consist of following processes:

'postgres', 'findy', 'sh', 'bash', 'start.sh'

'postgres' process correspond to the database server and 'findy' is the backend server. The rest of the processes are associated processes that involved in setting up the systems.

Send Rate	F statistic	P-Value	Numer. d.f.	Denom. d.f.	F critical value
5	49921.11	0	4	995	2.3809
10	66294.61	0	4	495	2.3899
20	3196.36	2.778038e-223	4	250	2.4078
50	668.54	3.431797e-81	4	100	2.4626

Table 24: ANOVA test for register in processes of decentralized Identity system

Send Rate	F statistic	P-Value	Numer. d.f.	Denom. d.f.	F critical value
5	1097.99	0	4	1070	2.3802
10	527.59	5.52106e-200	4	570	2.3876
20	46.53	7.30780e-41	4	323	2.3996
50	1292.88	9.825e-139	4	175	2.4223

Table 25: ANOVA test for login in processes of decentralized Identity system

The test results for the decentralized identity system are presented in Tables 24 and 25. These tables show the F statistic, p-value, degrees of freedom (numer. d.f. and denom. d.f.), and the critical F value for each send rate (5, 10, 20, and 50).

For the register process, all send rates resulted in extremely low p-values ($p < 0.05$), indicating strong evidence to reject the null hypothesis. The F statistic values were considerably high compared to the critical F values, further supporting the rejection of the null hypothesis.

Similarly, for the login process, all send rates yielded p-values below the significance level of 0.05, providing strong evidence to reject the null hypothesis. The F statistic values were also notably higher than the critical F values, reinforcing the rejection of the null hypothesis.

Based on these results, it can be concluded that there are significant differences in power consumption for the components within the decentralized identity system. The low p-values and high F statistic values indicate a strong association between the processes and power consumption.

4.2.3.1.3 Test result for federated identity system

In the implemented decentralized identity system, it consist of following processes:

'redis', 'mysqld', 'engine'

'redis' process correspond to the message broker, 'engine' is the backend server and 'mysqld' is the database.

Send Rate	F statistic	P-Value	Numer. d.f.	Denom. d.f.	F critical value
5	8775.58	0	2	612	3.0104
10	2802.24	1.633839e-210	2	313	3.0246
20	714.39	2.187029e-91	2	163	3.0515
50	95.73	2.259001e-31	2	75	3.1186

Table 26: ANOVA test for login in processes of federated Identity system

Send Rate	F statistic	P-Value	Numer. d.f.	Denom. d.f.	F critical value
5	82554.91	0	2	597	3.0108
10	9621.69	1.001845e-280	2	297	3.0262
20	2564	5.18154e-125	2	147	3.0576
50	400.77	2.69257e-44	2	57	3.1588

Table 27: ANOVA test for register in processes of federated Identity system

The ANOVA test results for the login and register processes are presented in Tables 26 and 27, respectively.

For the login process, the F statistic values were 8775.58, 2802.24, 714.39, and 95.73 for different send rates. The corresponding p-values were observed to be 0, 1.633839e-210, 2.187029e-91, and 2.259001e-31, all significantly smaller than the significance level of 0.05. This indicates strong evidence to reject the null hypothesis and accept the alternative hypothesis, confirming the presence of significant differences in power consumption among the

components. Additionally, the F critical value of 3.0104 for the send rate of 5 serves as a reference point for comparison.

Similarly, for the register process, the F statistic values were 82554.91, 9621.69, 2564, and 400.77, and the corresponding p-values were 0, 1.001845e-280, 5.18154e-125, and 2.69257e-44, all again smaller than 0.05. These results indicate significant differences in power consumption among the components for the register process as well. The F critical value of 3.0108 for the send rate of 5 serves as the threshold for significance comparison.

Overall, the ANOVA test results provide strong evidence of significant differences in power consumption among the components within the federated identity system for both the login and register processes. These findings suggest that the different processes contribute to varying power consumption levels, highlighting the importance of considering component-level power optimization strategies in the implementation of federated identity systems.

4.2.3.2 Tukey's HSD Pairwise test analysis

4.2.3.2.1 Test result for Centralized identity system

Send Rate Comparison	Statistic	p-value	Lower CI	Upper CI
5				
(Postgres - Python)	-1.827	0.000	-1.910	-1.743
(Python - Postgres)	1.827	0.000	1.743	1.910
10				
(Postgres - Python)	-3.142	0.000	-3.410	-2.875
(Python - Postgres)	3.142	0.000	2.875	3.410
20				
(Postgres - Python)	-5.045	0.000	-5.858	-4.231
(Python - Postgres)	5.045	0.000	4.231	5.858
50				
(Postgres - Python)	-6.977	0.000	-9.542	-4.412
(Python - Postgres)	6.977	0.000	4.412	9.542

Table 28: Tukey's HSD Pairwise Comparisons for Processes in Centralized system (Register)

Send Rate Comparison	Statistic	p-value	Lower CI	Upper CI
5				
(Postgres - Python)	-1.580	0.000	-1.618	-1.542
(Python - Postgres)	1.580	0.000	1.542	1.618
10				
(Postgres - Python)	-2.869	0.000	-2.924	-2.814
(Python - Postgres)	2.869	0.000	2.814	2.924
20				
(Postgres - Python)	-5.068	0.000	-5.150	-4.985
(Python - Postgres)	5.068	0.000	4.985	5.150
50				
(Postgres - Python)	-10.110	0.000	-10.312	-9.908
(Python - Postgres)	10.110	0.000	9.908	10.312

Table 29: Tukey's HSD Pairwise Comparisons for Processes in Centralized system (Login)

The provided test results in table 28 and 29 show the power consumption of different processes in a centralized identity system for the "Register" and "Login" scenarios, respectively.

For the "Register" scenario, the power consumption comparison reveals statistically significant differences between the "Postgres" and "Python" processes at different send rates. At a send rate of 5, the "Postgres" process consumes less power compared to "Python" (-1.827 vs. 1.827). This trend continues at higher send rates as well, with increasing differences in power consumption between the two processes.

Similarly, for the "Login" scenario, the power consumption comparison again demonstrates significant differences between the "Postgres" and "Python" processes. As the send rate

increases, the difference in power consumption between the two processes becomes more prominent.

In summary, the test results indicate that the power consumption of the "Postgres" process tends to be lower than that of the "Python" process in both the "Register" and "Login" scenarios.

4.2.3.2.2 Test result for decentralized identity system

Send Rate Comparison	Statistic	p-value	Lower CI	Upper CI
5				
(Postgres - Findy)	-21.840	0.000	-22.032	-21.647
(Findy - Postgres)	21.840	0.000	21.647	22.032
10				
(Postgres - Findy)	-41.320	0.000	-41.636	-41.003
(Findy - Postgres)	41.320	0.000	41.003	41.636
20				
(Postgres - Findy)	-63.250	0.000	-65.470	-61.030
(Findy - Postgres)	63.250	0.000	61.030	65.470
50				
(Postgres - Findy)	-106.716	0.000	-115.059	-98.374
(Findy - Postgres)	106.716	0.000	98.374	115.059

Table 30: Tukey's HSD Pairwise Comparisons for Processes in Decentralized system (Register)

Send Rate Comparison	Statistic	p-value	Lower CI	Upper CI
5				
(Postgres - Findy)	-0.697	0.000	-0.739	-0.655
(Findy - Postgres)	0.697	0.000	0.655	0.739
10				
(Postgres - Findy)	-1.345	0.000	-1.462	-1.228
(Findy - Postgres)	1.345	0.000	1.228	1.462
20				
(Postgres - Findy)	-5.473	0.000	-7.087	-3.858
(Findy - Postgres)	5.473	0.000	3.858	7.087
50				
(Postgres - Findy)	-13.638	0.000	-14.423	-12.853
(Findy - Postgres)	13.638	0.000	12.853	14.423

Table 31: Tukey's HSD Pairwise Comparisons for Processes in Decentralized system (Login)

The test results for the decentralized identity system are presented in Tables 30 and 31. These tables show the power consumption comparisons between the Postgres and Findy processes at different send rates for the Register and Login operations, respectively.

In the Register operation, at all send rates (5, 10, 20, and 50), the power consumption of the Findy process is significantly higher than that of the Postgres process. The statistical analysis reveals statistically significant differences with p-values of 0.000 for all comparisons. The power consumption difference ranges from -106.716 to 106.716, indicating substantial variability between the processes.

Similarly, in the Login operation, the Findy process also demonstrates significantly higher power consumption compared to the Postgres process across all send rates. The p-values are 0.000 for all comparisons, indicating highly significant differences. The power consumption difference ranges from -13.638 to 13.638.

These results suggest that the Findy process consistently consumes more power than the Postgres process in both the Register and Login operations of the decentralized identity system.

4.2.3.2.3 Test result for Federated identity system

Send Rate Comparison	Statistic	p-value	Lower CI	Upper CI
5				
(mysqld.log - engine.log)	7.651	0.000	7.470	7.832
(mysqld.log - redis)	30.101	0.000	29.920	30.282
(engine.log - mysqld.log)	-7.651	0.000	-7.832	-7.470
(engine.log - redis)	22.450	0.000	22.269	22.631
(redis - mysqld.log)	-30.101	0.000	-30.282	-29.920
(redis - engine.log)	-22.450	0.000	-22.631	-22.269
10				
(mysqld.log - engine.log)	65.707	0.000	64.414	66.999
(mysqld.log - redis)	66.137	0.000	64.845	67.430
(engine.log - mysqld.log)	-65.707	0.000	-66.999	-64.414
(engine.log - redis)	0.431	0.713	-0.862	1.723
(redis - mysqld.log)	-66.137	0.000	-67.430	-64.845
(redis - engine.log)	-0.431	0.713	-1.723	0.862
20				
(mysqld.log - engine.log)	110.916	0.000	106.666	115.165
(mysqld.log - redis)	111.678	0.000	107.428	115.927
(engine.log - mysqld.log)	-110.916	0.000	-115.165	-106.666
(engine.log - redis)	0.762	0.905	-3.487	5.011
(redis - mysqld.log)	-111.678	0.000	-115.927	-107.428
(redis - engine.log)	-0.762	0.905	-5.011	3.487
50				
(mysqld.log - engine.log)	186.450	0.000	168.088	204.813
(mysqld.log - redis)	187.735	0.000	169.372	206.098
(engine.log - mysqld.log)	-186.450	0.000	-204.813	-168.088
(engine.log - redis)	1.284	0.985	-17.078	19.647
(redis - mysqld.log)	-187.735	0.000	-206.098	-169.372
(redis - engine.log)	-1.284	0.985	-19.647	17.078

Table 32: Tukey’s HSD Pairwise Comparisons for Processes in Federated system (Register)

The Tukey’s HSD pairwise comparisons were performed to analyze the power consumption of different processes in the Federated identity system. The results for the ”Register” operation are summarized in Table 32, while the results for the ”Login” operation are summarized in Table 33.

In the ”Register” operation, at a send rate of 5, the power consumption difference between ”mysqld.log” and ”engine.log” was 7.651, and the p-value was significant ($p < 0.001$), indicating a statistically significant difference. Similarly, significant differences were observed for other comparisons. As the send rate increased to 10, 20, and 50, the power consumption differences also increased, and all the differences were statistically significant ($p < 0.001$) except for one non-significant difference at a send rate of 10 between ”engine.log” and ”redis”.

Send Rate Comparison	Statistic	p-value	Lower CI	Upper CI
5				
(mysqld.log - engine.log)	-29.297	0.000	-29.840	-28.754
(mysqld.log - redis)	-22.090	0.000	-22.634	-21.546
(engine.log - mysqld.log)	29.297	0.000	28.754	29.840
(engine.log - redis)	7.207	0.000	6.658	7.756
(redis - mysqld.log)	22.090	0.000	21.546	22.634
(redis - engine.log)	-7.207	0.000	-7.756	-6.658
10				
(mysqld.log - engine.log)	-52.590	0.000	-54.258	-50.922
(mysqld.log - redis)	-31.130	0.000	-32.806	-29.453
(engine.log - mysqld.log)	52.590	0.000	50.922	54.258
(engine.log - redis)	21.460	0.000	19.750	23.171
(redis - mysqld.log)	31.130	0.000	29.453	32.806
(redis - engine.log)	-21.460	0.000	-23.171	-19.750
20				
(mysqld.log - engine.log)	-87.854	0.000	-93.382	-82.326
(mysqld.log - redis)	-48.821	0.000	-54.407	-43.235
(engine.log - mysqld.log)	87.854	0.000	82.326	93.382
(engine.log - redis)	39.033	0.000	33.237	44.830
(redis - mysqld.log)	48.821	0.000	43.235	54.407
(redis - engine.log)	-39.033	0.000	-44.830	-33.237
50				
(mysqld.log - engine.log)	-128.296	0.000	-150.873	-105.719
(mysqld.log - redis)	-77.982	0.000	-101.138	-54.826
(engine.log - mysqld.log)	128.296	0.000	105.719	150.873
(engine.log - redis)	50.314	0.000	25.635	74.993
(redis - mysqld.log)	77.982	0.000	54.826	101.138
(redis - engine.log)	-50.314	0.000	-74.993	-25.635

Table 33: Tukey's HSD Pairwise Comparisons for Processes in Federated system (Login)

In the "Login" operation, similar patterns were observed. At a send rate of 5, the power consumption difference between "mysqld.log" and "engine.log" was -29.297, indicating a higher power consumption for "engine.log". All the differences were statistically significant ($p < 0.001$) for the comparisons at all send rates. As the send rate increased, the power consumption differences also increased, showing the impact of the send rate on power consumption.

These results indicate that different processes have varying power consumption levels in the Federated identity system for both "Register" and "Login" operations.

5 CONCLUSION

The research addresses three research questions related to power consumption in different models of identity systems. In response to RQ1, statistical tests reveal significant differences in power consumption among the three implemented identity systems. Contrary to the initial assumption based on previous research, decentralized identity systems do not always consume the most power. Moving on to RQ2, power consumption data is collected for different levels of user activity, specifically send rates. The results indicate a positive correlation between send rate and power consumption across all operations (register, login). This finding aligns with the initial assumption that higher send rates have a greater impact on power consumption. Finally, RQ3 investigates the components within the identity systems that consume the most power. The collected data show that backend components, rather than database components, consistently consume the most power across all operations and send rates, supporting the initial assumption that backend components are the primary power consumers.

5.1 Summary of Findings

5.1.1 RQ1: Difference in power consumption between different model of identity systems

From the statistical test results, it is clear that there are significant difference in power consumption of three implemented identity systems. The starting assumption is that decentralized identity systems will consumption most power based on the result from previous work by (Sedlmeir et al. 2020), however the results shown that it is not hard truth that decentralized identity systems will always consume the most power.

5.1.2 RQ2: Power consumption of different type of identity management system vary with different levels of user activity

For RQ2, power consumption data of different send rate (5, 10, 20, 50) are collected. Across all operations (register, login), the data shown that power consumption does increase as the send rate increase and this suggested there is positives correlation between the both variables and this matched with the starting assumption that send rate has a positive impact on the power consumption.

5.1.3 RQ3: Components that consume the most power in identity management system

For RQ3, the power consumption data are collected from each processes within the identity systems. The processes within the identity system can mainly divided into backend and database components. The test result shown that backend components consume the most power across all operations and all send rate and this is in line with the starting assumption that backend components is the one that consume most power.

5.2 Discussion of Result

5.2.1 contribution of the study

The findings from RQ1 offer valuable insights into the power consumption of blockchain systems compared to their non-blockchain counterparts. Contrary to assumptions, the study reveals that blockchain-based systems do not inherently consume more power. This highlights the need to avoid making blanket assumptions about the power consumption of blockchain technologies and underscores the significance of conducting comprehensive evaluations.

The study's contribution extends beyond the specific research questions, as it introduces a practical and open-source framework for measuring power consumption within Docker containers. This framework can be readily adapted and used by researchers and practitioners to evaluate the energy efficiency of various software and services hosted in containerized environments. By offering this versatile tool, the study empowers the broader community to conduct power consumption assessments, fostering greater awareness of sustainable practices in software development and deployment.

The research study's sustainability impact can be linked to the United Nations Sustainable Development Goal (SDG) 7: Affordable and Clean Energy. By exploring ways to optimize power consumption in identity systems, the study aligns with the broader global goal of ensuring access to affordable, reliable, sustainable, and modern energy for all. Implementing energy-efficient systems in the digital landscape, such as identity management systems, contributes to reducing the environmental impact and promoting sustainable development.

Overall, this research study advances the understanding of power consumption in identity systems, offers practical tools for assessing energy usage, and contributes to the broader goal of achieving clean and sustainable energy practices as outlined in SDG 7.

5.2.2 Limitation of the experiment

Firstly, it is important to note that the experiments were conducted on a single laptop, which may have limited the scalability and generalizability of the findings. Power consumption measurements on a larger scale, involving multiple devices or distributed systems, would provide a more comprehensive understanding of the energy requirements of blockchain-based decentralized identity systems. Additionally, the use of a single laptop may have introduced certain confounding factors, such as variations in hardware specifications and operating system settings, which could influence power consumption measurements. Furthermore, the experiments were performed within the constraints of the available computational resources, potentially limiting the complexity and size of the datasets used for analysis. Despite these limitations, the study still offers valuable insights into the relative power consumption of different identity models, shedding light on the energy efficiency considerations in the realm of decentralized identity systems.

5.2.3 Future work

Future work for this study could involve expanding the experiments to a distributed system on a larger scale. By running the experiments in a distributed environment, the power consumption of blockchain-based decentralized identity systems can be evaluated under real-world conditions. Additionally, exploring the impact of implementing different blockchain technologies could provide valuable insights into the energy efficiency of specific blockchain platforms and their suitability for decentralized identity systems. Furthermore, increasing the sample size for the three identity models would enhance the statistical robustness of the findings and allow for a more comprehensive analysis of the power consumption variations. These potential future research directions would contribute to a deeper understanding of the energy implications and scalability of blockchain-based decentralized identity systems.

REFERENCES

- Acar, Hayri et al. (2016). “The impact of source code in software on power consumption”. In: *International Journal of Electronic Business Management* 14, pp. 42–52.
- Alshahrani, Hani et al. (2023). “Sustainability in Blockchain: A Systematic Literature Review on Scalability and Power Consumption Issues”. In: *Energies* 16.3, p. 1510.
- Bertino, Elisa and Kenji Takahashi (2010). *Identity management: Concepts, technologies, and systems*. Artech House.
- Cao, Yuan and Lin Yang (2010). “A survey of identity management technology”. In: *2010 IEEE International Conference on Information Theory and Information Security*. IEEE, pp. 287–293.
- Dib, Omar and Khalifa Toumi (2020). “Decentralized identity systems: Architecture, challenges, solutions and future directions”. In: *Annals of Emerging Technologies in Computing (AETiC), Print ISSN*, pp. 2516–0281.
- Ergasheva, Shokhista et al. (2020). “Metrics of energy consumption in software systems: a systematic literature review”. In: *IOP Conference Series: Earth and Environmental Science*. Vol. 431. 1. IOP Publishing, p. 012051.
- Grassi, Paul, Michael Garcia, James Fenton, et al. (2020). “NIST Digital Identity Guidelines”. In: <https://csrc.nist.gov/publications/detail/sp/800-63/3/final>.
- Hindle, Abram (2012). “Green mining: A methodology of relating software change to power consumption”. In: *2012 9th IEEE Working Conference on Mining Software Repositories (MSR)*. IEEE, pp. 78–87.
- Jensen, Jostein (2012). “Federated identity management challenges”. In: *2012 Seventh International Conference on Availability, Reliability and Security*. IEEE, pp. 230–235.
- Kubach, Michael et al. (2020). “Self-sovereign and Decentralized identity as the future of identity management?” In: *Open Identity Summit 2020*.
- Ah-Lian, Kor et al. (2019). “Education in green ICT and control of smart systems: A first hand experience from the International PERCCOM masters programme”. In: *IFAC-PapersOnLine* 52.9, pp. 1–8.
- Lim, Shu Yun et al. (2018). “Blockchain technology the identity management and authentication service disruptor: a survey”. In: *International Journal on Advanced Science, Engineering and Information Technology* 8.4-2, pp. 1735–1745.
- Naik, Nitin and Paul Jenkins (2017). “Securing digital identities in the cloud by selecting an apposite Federated Identity Management from SAML, OAuth and OpenID Connect”. In: *2017 11th International Conference on Research Challenges in Information Science (RCIS)*. IEEE, pp. 163–174.

- Noureddine, Adel (2022). “PowerJoular and JoularJX: multi-platform software power monitoring tools”. In: *2022 18th International Conference on Intelligent Environments (IE)*. IEEE, pp. 1–4.
- Pöhn, Daniela and Wolfgang Hommel (2020). “An overview of limitations and approaches in identity management”. In: *Proceedings of the 15th International Conference on Availability, Reliability and Security*, pp. 1–10.
- Preukschat, Alex, Drummond Reed, and Doc Searls (2021). “The three models of digital identity”. In: *Self-sovereign identity: Decentralized Digital Identity and verifiable credentials*. Manning, pp. 9–11.
- Schinckus, Christophe (2020). “The good, the bad and the ugly: An overview of the sustainability of blockchain technology”. In: *Energy Research & Social Science* 69, p. 101614.
- Sedlmeir, Johannes et al. (2020). “The energy consumption of blockchain technology: Beyond myth”. In: *Business & Information Systems Engineering* 62.6, pp. 599–608.
- Smith, Samuel M and Dmitry Khovratovich (2016). “Identity system essentials”. In: *Ev-ernym*, Mar 29, p. 16.
- Stokkink, Quinten et al. (2021). “A truly self-sovereign identity system”. In: *2021 IEEE 46th Conference on Local Computer Networks (LCN)*. IEEE, pp. 1–8.

A APPENDIX: SOURCE CODE

The source code used in this study are hosted on Github with the following address:
https://github.com/michiboo/IDM_PCM_Thesis.git