

UNIVERSITY OF BAHRAIN
College of Information Technology



**A BLOCKCHAIN-BASED INTELLIGENT AUTOMATION
SOLUTION TO COMBAT DIGITAL FRAUD AND DATA
LEAKAGE IN B2B E-COMMERCE**

A Thesis Submitted in Partial Fulfilment of the Requirements for the Doctor of
Philosophy Degree in Computing and Information Sciences

Submitted by:

Yaqoob Salman Mohamed Al-Slais

20190179

Supervised by:

Dr. Mazen Ali

(Associate Professor)

Kingdom of Bahrain

June 2024

APPROVAL

University of Bahrain
Deanship of Graduate Studies
And scientific Research



جامعة البحرين
عمادة الدراسات العليا والبحث العلمي

Dissertation Deposit Form

Author Name:	Yaqoob Salman Mohamed Al-Slais		
Email:	ysalslais@uob.edu.bh	Contact No:	38887424
College:	Information Technology		
Department:	Information Systems		
Thesis Title:	A BLOCKCHAIN-BASED INTELLIGENT AUTOMATION SOLUTION TO COMBAT DIGITAL FRAUD AND DATA LEAKAGE IN B2B E-COMMERCE		
Year:	2024		

Author's Declaration: I agree to the following conditions:

The above thesis will be available (in the University of Bahrain library and externally) and reproduced as necessary at the discretion of the University of Bahrain. It may also be digitized by the University of Bahrain and made available via the university's repository or via other parties on the Internet.

This condition shall apply to ALL copies including electronic copies.

The above thesis has been provided on the understanding that it is copyright protected material and that no quotation from this thesis may be published without proper acknowledgement.

Signature of the Author:	YAQOOB SALMAN ALSLAIS	Date:	23-Oct-24
--------------------------	-----------------------	-------	-----------

10 | Approved at the University Council 6th meeting, 2013
In accordance with Decision No. 736/2013 dated 14/3/2013
Modified by University Council Decision No. 2043/2013 dated 20/11/2013.

DECLARATION

DECLARATION

I declare that this work is the result of my own investigation and that it has not already been accepted in substance for any degree, nor is it currently submitted for any degree.

Signed:  Date: 22/06/2024

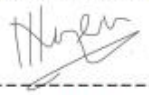
Yaqoob Salman Mohamed Al-Slais (Candidate)

The Thesis defense committee considers the thesis titled:

A Blockchain-based Intelligent Automation Solution to Combat Digital Fraud and Data Leakage in B2B E-Commerce

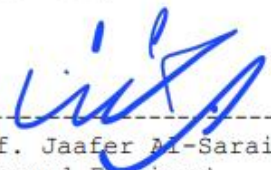
Submitted by Yaqoob Salman Mohamed Al-Slais to the College of Information Technology is satisfactory and acceptable for the PhD. in Computing and Information Sciences.

The Defense Committee:



Dr. Mazen Ali
(Supervisor)

Department of Information Systems
University of Bahrain



Prof. Jaafer Al-Sarairah
(External Examiner)

Princess Sumaya University
for Technology



Dr. Haroun Alryalat
(Internal Examiner)

Department of Information Systems
University of Bahrain

ABSTRACT

As the 4th Industrial Revolution advances and the shift towards a digital workforce intensifies, automation and cybersecurity have become more critical than ever for business enterprises. Robotic Process Automation (RPA) and its AI-driven counterpart, Intelligent Automation (IA), are soft targets for cyberattacks as many of these bots handle clients' sensitive personal data. Many current IA applications suffer from poor logging and auditing practices, as attackers can easily manipulate or delete activity logs to cover their tracks after exploiting the IA bot. This thesis proposes a Blockchain-based Intelligent Automation (BCIA) framework to combat digital fraud and data leakage. By integrating Blockchain with Intelligent Automation, the BCIA Bot Framework automates invoice processing tasks by utilizing Optical Character Recognition (OCR) while securing the bot's activity logs and payment processing receipt evidence onto a BC ledger. The Design Science Research Methodology (DSRM) was applied to guide the development and evaluation of the BCIA framework. This approach ensured a systematic exploration and implementation of the technology. A scenario-based strategy was adopted to simulate real-world applications of the framework for small businesses automating invoice processing in two critical areas of combating data leakage and digital fraud. The BCIA bot successfully stored the activity logs and payment process receipt evidence on the BC at different transaction volumes representing small enterprises (50 and 100 invoices), medium-sized enterprises (500 invoices), and large enterprises (1000 invoices). As for the security framework aspect, despite the BCIA Bot Framework utilizing a public blockchain and a cloud service infrastructure, while the Block-DEF and Blockchain Forensic Logging (BCFL) frameworks utilized private blockchains, the BCIA Bot Framework was able to create a solid foundation for public blockchain IA bot development that provides secure evidence creation and retrieval through a cloud ecosystem. In comparing the system performance metrics of three BC testnets (Sepolia ETH, Solana Devnet, and BNB Chain), which include network throughput, block sizes, transaction fees, transaction latency, and finality time on the BCIA bot, BNB Chain delivered the most efficient performance and cost-effective solutions for small, medium, and large businesses. The BNB-BCIA bot achieved a high average throughput of 47 transactions per minute, an average finality time of 9.3 seconds, and low transaction costs, achieving \$USD 4.20 to process 1000 invoices.

Table of Contents

APPROVAL	A
DECLARATION.....	B
ABSTRACT	C
Table of Figures.....	H
List of Tables	I
ACKNOWLEDGMENTS	K
DEDICATION	L
PUBLICATIONS	M
LIST OF ABBREVIATIONS.....	N
Chapter One: Introduction	2
1.1 Overview:.....	2
1.2 The Evolution of Robotic Processing Automation	3
1.3 Digital Fraud and Data Leakage	4
1.4 Research Problem	7
1.5 Research Objectives and Research Questions	8
1.6 Research Contribution to the Field	10
1.7 Research Method	11
1.8 Thesis Outline	12
1.9 Summary	12
Chapter 2 Literature Review	14

2.1 Overview.....	14
2.2 Blockchain Technology	14
2.3 Consensus algorithms	17
2.3.1 Proof of Work (PoW).....	18
2.3.2 Proof of Stake (PoS).....	18
2.3.3 Delegated Proof of Stake (DPoS).....	19
2.3.4 Proof of Elapsed Time (PoET)	19
2.3.5 Practical Byzantine Fault Tolerance (PBFT).....	20
2.3.6 Proof of Authority (PoA).....	20
2.3.7 Proof of History (PoH)	20
2.3.8 Proof of Stake Authority (PoSA).....	21
2.3.9 Proof of Capacity (PoC)	21
2.3.10 Proof of Importance (PoI).....	22
2.3.11 Raft.....	22
2.3.12 Comparison of Consensus Algorithms	23
2.4 Testnets.....	25
2.5 Security Challenges in RPA and IA.....	26
2.5.1 SLR METHODOLOGY	30
2.5.2 Inclusion/ exclusion criteria.....	32
2.5.3 Snowballing	32
2.6 Results.....	34
2.6.1 Logging and Auditing	37

2.6.2 Access Control	38
2.6.3 Security Standards and Compliance	39
2.6.4 Network Security	39
2.6.5 Blockchain Security Vulnerabilities	39
2.6.6 AI Security Vulnerabilities	39
2.6.7 Encryption.....	40
2.6.8 Lack of Testing	40
2.7 RPA and IA for Invoice Processing	40
2.8 RPA, IA, and Blockchain Use Cases	42
2.9 Blockchain-based Invoice Processing	44
2.10 Blockchain for Digital Forensics evidence creation and storage:	45
2.10.1 Block-DEF	45
2.10.2 BCFL Logging Framework	46
2.11 Summary	47
Chapter 3 Methodology	49
3.1 Overview.....	49
3.2 Design Science Research Methodology	49
3.3 Problem Identification and Motivation.....	53
3.4 Design and Development: The BCIA Bot Framework.....	54
3.4.1 Receiving incoming invoices.....	56
3.4.2 Invoice Data Extraction	58
3.4.3 Invoice data validation and approval.....	59

3.4.4 Coding and posting	59
3.4.5 Payment processing	61
3.4.6 Adding Log entries from File to BC	62
3.4.7 Logging payment receipts onto the BC	64
3.5 BCIA Security Architecture	68
3.6 Summary	69
Chapter 4 Experimental Results and Discussion	71
4.1 Overview	71
4.2 Testnets And Performance Metrics Used In The Experiment	71
4.3 Dataset and Experiment Setup:	74
4.4 Experiment Metrics Equations:	75
4.5 Experiment 1: ETH-BCIA bot	78
4.6 Experiment 2: SOL-BCIA Bot	84
4.7 Experiment 3: BNB-BCIA Bot	89
4.8 Cost Effectiveness	95
Chapter 5 Conclusions and Recommendations	102
5.1 Conclusions	102
5.2 Future Work and Recommendations	106
Appendix	108
References:	109

Table of Figures

Figure 1 Initial SLR and Snowballing Process	30
Figure 2 Number of Publication types by year (Jan 2017- May 2022)	34
Figure 3 DSRM Process Model (Peffer et al. 2007)	50
Figure 4 BCIA Bot Framework DSRM Process Model	52
Figure 5 Data Leakage Scenario	53
Figure 6 Digital Fraud Scenario.....	54
Figure 7 The BCIA Bot Framework	55
Figure 8 Example of BCIA bot activity log.....	62
Figure 9 ETH-BCIA bot at 50 invoices	79
Figure 10 ETH-BCIA at 100 invoices	80
Figure 11 ETH-BCIA bot performance at 500 invoices	82
Figure 12 ETH-BCIA bot performance at 1000 invoices.....	83
Figure 13 SOL-BCIA bot performance at 50 invoices	85
Figure 14 SOL-BCIA bot performance at 100 invoices	86
Figure 15 SOL-BCIA bot performance at 500 invoices	87
Figure 16 SOL-BCIA bot performance at 1000 invoices	88
Figure 17 BNB-BCIA bot performance at 50 invoices	91
Figure 18 BNB-BCIA bot performance at 100 invoices	92
Figure 19 BNB-BCIA bot performance at 500 invoices	93
Figure 20 BNB-BCIA bot performance at 1000 invoices	94
Figure 21 An example of Ethereum gas fees fluctuation over a 7-day period (Ethereum Gas Fees , 2024).....	96

List of Tables

Table 1 Comparison of five Consensus Algorithms (Mingxiou et al.,2017).....	24
Table 2 Blockchain Testnets and their corresponding consensus algorithms.....	26
Table 3 RPA security challenges and their categories according to Previous Literature	28
Table 4 Initial Search Terms by Digital Library	31
Table 5 Frequency of SLR-selected publications	33
Table 6 RPA and IA Security Challenges based on SLR.....	35
Table 7 Invoice data extraction fields	58
Table 8 Comparison between Block-DEF, BCFL logging and BCIA Frameworks	69
Table 9 Average block generation time and Finality time for Solana, BNB Chain, and ETH	73
Table 10 Testnet programming languages and execution environments	75
Table 11 ETH-BCIA bot performance at 50 invoices.....	78
Table 12 ETH-BCIA bot performance at 100 invoices	79
Table 13 ETH-BCIA bot performance at 500 invoices	81
Table 14 ETH-BCIA bot performance at 1000 invoices	82
Table 15 ETH-BCIA bots transaction fees and Finality Times	84
Table 16 SOL-BCIA Bot performance at 50 invoices	85
Table 17 SOL-BCIA Bot performance for 100 invoices	85
Table 18 SOL-BCIA bot's performance at 500 invoices	86
Table 19 SOL-BCIA bot performance at 1000 invoices.....	88
Table 20 Total transaction fees and Finality Time averages for SOL-BCIA Bot	89
Table 21 BNB-BCIA Bot performance at 50 invoices	91
Table 22 BNB-BCIA Bot performance at 100 invoices	91
Table 23 BNB-BCIA Bot performance for 500 invoices.....	92

Table 24 BNB-BCIA Bot performance at 1000 invoices	93
Table 25 Transaction fees and Finality Times for BNB-BCIA Bot.....	95
Table 26 Average Transaction fees in cryptocurrency and \$USD (7-14 April 2024)	96
Table 27 ETH-BCIA bot's ETH transaction fees to \$USD conversion	97
Table 28 SOL-BCIA Bot's SOL transaction fee conversion to \$USD.....	97
Table 29 BNB-BCIA Bot transaction fee conversion to \$USD.....	97
Table 30 Comparison of BCFL and BCIA Bot network throughput and latency....	99
Table 31 Comparison between Block-DEF and BCIA Framework.....	100

ACKNOWLEDGMENTS

I sincerely thank my supervisor, Dr. Mazen Ali, for his support throughout my academic journey. I would also like to express my eternal gratitude to my colleagues at the University of Bahrain for their support and encouragement. Finally, this work would have never have become a reality without the loving support of my mother, wife, and children.

DEDICATION

I dedicate this work to my late father, Salman Mohamed Yaqoob Al-Slais, my eldest brother, Luqman, and the many teachers and father figures throughout my academic journey and in life.

PUBLICATIONS

This thesis is based on the following published research paper written jointly with my supervisor, Dr. Mazen Ali:

CONFERENCE PAPERS

1. Al-Slais, Y., & Ali, M. (2023, January). Robotic Process Automation and Intelligent Automation Security Challenges: A Review. In *2023 International Conference On Cyber Management And Engineering (CyMaEn)* (pp. 71-77). IEEE.

The following publications were completed as part of the PhD program in Computing and Information Sciences at the University of Bahrain.

1. (Conference Paper) Al-Slais, Y. (2020, December). Privacy engineering methodologies: A survey. In *2020 International Conference on Innovation and Intelligence for Informatics, Computing and Technologies (3ICT)* (pp. 1-6). IEEE.
2. (Journal Article) Al-Slais, Y., and El-Medany, W. M (2022). User Centric Adaptive Password Policies to Combat Password Fatigue. *Int. Arab J. Inf. Technol.* , 19(1), 55-62.
3. (Conference Paper) Al-Slais, Y.S (2021) Towards a comprehensive self-admitted technical debt extraction technique from source code comments. In *3rd Smart Cities Symposium (SCS 2020)* , 2021, pp. 109-114). IET.

LIST OF ABBREVIATIONS

4IR	Fourth Industrial Revolution
ABI	Application Binary Interface
AI	Artificial Intelligence
API	Application Programming Interface
ASIC	Application-Specific Integrated Circuit
AWS	Amazon Web Services
B2B	Business-to-Business
BaaS	Blockchain as a Service
BC	Blockchain
BCFL	Blockchain Forensic Logging
BCIA	Blockchain-based Intelligent Automation
BNB	Abbreviation for ‘Build and Build’
BPM	Business Process Management
CA	Consensus Algorithm
DApps	Decentralized Apps
DLT	Distributed Ledger Technology
DoS	Denial of Service Attack
DPoS	Delegated Proof of Stake

DSRM	Design Science Research Methodology
EAI	Enterprise Application Integration
ETH	Ethereum cryptocurrency
EVM	Ethereum Virtual Machine
GDPR	General Data Protection Regulation
GPU	Graphical Processing Unit
HA	Hyper Automation
IA	Intelligent Automation
IaaS	Infrastructure-as-a-Service
IT	Information Technology
MitM	Man-in-the-Middle Attack
ML	Machine Learning
NFT	Non-Fungible Token
OCR	Optical Character Recognition
PBFT	Practical Byzantine Fault Tolerance
PDF	Portable Document Format
PII	Personally Identifiable Information
PKI	Public / Private Key Infrastructure
PoA	Proof of Authority
PoC	Proof of Capacity

PoET	Proof of Elapsed Time
PoH	Proof of History
PoI	Proof of Importance
PoS	Proof of Stake
PoSA	Proof of Stake Authority
PoW	Proof of Work
RBAC	Role-Based Access Control
RPA	Robotic Process Automation
RQ	Research Question
SaaS	Software-as-a-Service
SC	Smart Contracts
SLR	Systematic Literature Review
SME	Small and Medium Enterprise
TPS	Transactions Per Second
UI	User Interface

Chapter One: Introduction

1.1 Overview:

From the invention of water wheels in the first century BC, to the industrial revolution of the 17th and 18th centuries to the rise of computer usage and robotics in the 20th and 21st centuries, Automation has always been the core characteristic and hallmark of every human technological advance and industrial revolution. Automation and its levels are determined by comparing human and machine workloads, where a device or system accomplishes tasks fully or partially, previously carried out by a human operator exclusively (Parasuraman et al., 2000). Robotic Process Automation (RPA) has rapidly grown into a multibillion-dollar industry and a core component of what is now labeled the fourth industrial revolution. The term RPA itself emerged in the early 21st century, but some experts argue (Madakam et al., 2019; Willcocks, 2020; Willcocks et al., 2017) that its first manifestations occurred in the late 20th century.

RPA attempts to mimic human office tasks or “swivel chair” tasks that require multiple applications to complete repetitive tasks. The RPA software bots are considered the face of the new digital workforce, utilizing rule-based logic and combining user interface (UI) interaction recordings with application programming interfaces (API) to perform repetitive and time-consuming tasks across enterprise and productivity applications. Businesses benefit from RPA bots, as they lift the burden off human employees from doing repetitive work, which leads to inevitable human errors, and lets them focus on more creative problem-solving roles in their enterprises.

RPAs consist of three main components: Robots, both assisted and unassisted, an Orchestrator, and a software Studio (Choi et al., 2021). An assisted RPA bot is a bot that cannot complete tasks without human intervention. In contrast, unassisted RPAs are fully automated and require no human intervention (*Security Risks in Robotic Process Automation (RPA)*, 2022). Cognitive RPAs are RPAs that utilize ML for decision-making and self-improvement.

There are three main approaches to building RPAs: Demonstration, Natural Language Processing (NLP), and Rewarding (Chakraborti et al., 2020). The demonstration approach automates tasks through human observation or by processing the software's activity logs. The NLP and Rewarding approaches utilize machine learning algorithms to learn the business process. NLP achieves feature extraction by training the support vector machine and its rewarding scheme and applying reinforcement learning algorithms to reward RPAs and train them to improve performance over time. Both Demonstration and NLP approaches require some level of human input and intervention, whereas Rewarding, sometimes referred to as RPA 2.0, seeks to remove human-reliant training.

1.2 The Evolution of Robotic Processing Automation

As computers entered the workplace in the late 1960s and more so over the following two decades, they helped increase employee productivity in leaps and bounds as new applications were developed, making many once-manually laborious office work obsolete and gone within a click of a mouse. However, as desktop applications became increasingly complex and information systems that use a myriad of different applications became commonplace within business enterprises and organizations, employees found themselves doing repetitive tasks on the information systems,

which led to many human errors and wasted opportunities for employees to do more work that rely on creative problem-solving skills.

The first iteration of business process automation was macros, which recorded simple keystroke sequences to complete repetitive tasks or create complex scripts to process data automatically and generate reports (Rybinski & Schöler, 2022). This was followed by the rise of batch processing frameworks, which allowed developers to schedule and execute automated tasks at pre-defined intervals. RPAs created a fundamental shift where regular business personnel could now create automation bots with no or low-code experience through screen recording and workflow orchestration engines (La Rosa et al., 2016). In addition, Intelligent Automation (IA) utilized Artificial Intelligence (AI) and Machine Learning (ML) to overcome the rigidness of RPA bots with their rule-based structures and to make IA bots autonomous without human intervention (Baviskar et al., 2021). Finally, Hyper Automation (HA) combines all the automation technologies, methodologies, and tools mentioned above to streamline and comprehensively augment an organization's business processes (IBM Blog, 2021).

1.3 Digital Fraud and Data Leakage

Data Leakage and Digital Fraud are considered the two major risks facing RPA security today, according to Gartner (*Four Steps to Ensure Robotic Process Automation Security* | Gartner, 2022). Data leakage is a cybercrime where malicious actors illegally access and leak customers' personal data or an RPA bot's security credentials to third parties. Digital fraud, in an RPA context, is the illegal access of RPA bot credentials to commit monetary fraud by creating fraudulent transactions or siphoning funds to third-party funds under the guise of the RPA bot. RPAs do not

have the autonomy to commit malicious acts, so an attacker would most likely reprogram the bot to commit the crimes (Guidehouse, 2021).

In Kosi's work (Kosi, 2019), there are three primary cybersecurity risks in RPA systems: Firstly, Abuse of Privileged system access, where a malicious actor, such as a hacker or insider threat, gains access to a bot developer or manager's credentials and exploits the bot to leak data. Insider threats, e.g. disgruntled employees or the bot developer, can alter the original programming of the bot to sabotage and erase sensitive data, halt and delay automated business processes, or steal intellectual property. Secondly, Data Leakage can occur due to bot developers' poor security awareness and malpractices, such as allowing the bot to upload and access sensitive private data to a publicly accessible database on the web without encryption or failing to secure data transmissions from one location to another. Finally, Virtual Machine (VM) vulnerabilities involve hackers and other malicious actors exploiting security flaws to remotely access the VM or create service outages due to unplanned system upgrades and network maintenance.

According to Gartner's 2023 Peer Insight engine (Gartner, 2023), which compared the security features of the Top five RPA vendors, Logging was one of the pain points in the RPA vendors, as only Automation Anywhere and Blue Prism (BP) had built-in Audit logs. BP utilizes automatic logging and creates immutable and nonrepudiation logs, preventing log interference. In contrast, Automation Anywhere's logs have a manual recording approach that humans oversee, creating a heightened potential for human errors and log manipulation. Syed et al. listed in their publication (Syed et al., 2020) multiple approaches to securing RPA bots by enforcing thorough and robust logging and clear audit trails, complying to industry security standards, and applying rigorous access control policies. Researchers believe BC and Smart Contracts (SC) can provide feasible and practical solutions to

mitigate the risk of Data Leakage through immutable ledgers and data hashing techniques. For example, Forment et al. (Forment et al., 2018) proposed using SCs to automate legal actions whenever data collection or data usage irregularities are detected in a Moodle Learning Management System's (LMS) learning analytics systems to ensure data protection for their students. Amo et al. (Amo et al., 2019) proposed a similar approach by developing a Personal Data Broker plugin in Moodle LMS that utilized SCs to secure and protect students' activity logs and data automatically.

Another security risk is operating with outdated back-end systems in Intelligent Automation (IA) bots. Many organizations fail to consistently update their underlying systems, creating unwanted security gaps and vulnerabilities. Jurmu's work also described how a malicious actor could exploit an RPA bot through cyberattacks such as Denial of Service (DoS), SQL injection attacks on the systems's databases, or Man-in-the-Middle (MitM) attacks (Jurmu, 2019). Hartikainen et al. (Hartikainen et al., 2022) stressed that RPA maintenance in large enterprises is not up to security policy requirements and encouraged the use of security monitoring tools to detect and react to any security risks.

A systematic mapping study by (Enriquez et al., 2020) through a comprehensive analysis of 54 publications found limited research proposals for testing RPAs in a testing environment before deployment in a production environment. Their findings underlined a general lack of literature regarding RPA security risks and how to mitigate them, which aligned with the findings of (Syed et al., 2020)'s work, thus highlighting the importance of research in this nascent area of cybersecurity due to the currently limited amounts of studies and research tackling this essential yet overlooked problem.

Nokod Security in their report titled: (Low Code/No Code and RPA: Rewards and Risks, 2023), highlighted several cases of security breaches in RPA systems. One example is when a hospital operations manager developed an application for doctors which sent prescriptions to the hospital pharmacy. However, this sensitive data was accessible by hospital administrators through the RPA bot, who then could leak or tamper with the prescriptions. Another example is when an HR team at an organization created an RPA bot to collect employee T-shirt sizes for a company giveaway. However, in the backend, the bot accessed the full employee table in the company's database that held Personally Identifiable Information (PII) such as addresses, salary, and family members. A simple database query manipulation would have given unauthorized access to all the PII.

Many previous works restricted their findings and solutions to a more managerial and business-oriented approach (Enriquez et al., 2020; Hong et al., 2023; Kurylets & Goranin, 2023; Murugappan & Sree Kala, 2022; Syed et al., 2020). Thus, limited research and resources are discussing RPA and IA security challenges from a technical perspective. To address this gap in the scientific literature, the researcher proposes a Blockchain-based Intelligent Automation (BCIA) bot framework aimed at combating the threats of digital fraud and data leakage.

1.4 Research Problem

RPA and IA solutions are widely seen as practical and effective tools in combating digital fraud. Many banks and other financial institutions leverage RPA's rule-based approach to minimize human error and limit fraudulent transactions. IA's ML capabilities can also analyze historical data, identify transaction anomalies, and alert their organizations. However, many organizations grossly overlook RPA and IA

security issues, making them an easy target for exploitation, particularly in digital fraud and data leakage.

For example, disgruntled employees and malicious actors can illegally access RPA and IA bots by using the bots' login credentials as a gateway to access the organization's sensitive data, then create fraudulent transactions and process fake invoices under the guise of the RPA or IA bot. Malicious RPA developers can develop backdoors to facilitate the leaking of private and sensitive data to third parties. RPA and IA bots that are poorly maintained and not updated frequently to reflect the changes in the business processes are also vulnerable to exploitation and data leakage.

After committing their crime, malicious actors can modify or delete forensic evidence of their wrongdoing by manipulating and falsifying the RPA and IA bots' activity logs. Digital fraud and data leakage can cause hefty financial damage to small and medium-sized enterprises (SMEs) and large enterprises.

One of the most effective means to deter and detect committed cybercrimes is through thorough and robust logging practices. However, most propriety RPA software vendors have no security measures to ensure logs are not tampered with (Gartner, 2023). Thus, cybercriminals can easily manipulate or delete activity logs and destroy all the evidence required to launch a digital forensic investigation. Therefore, there is a critical need to find solutions to combat digital fraud and data leakage in B2B e-commerce that stem from the exploitation of RPA and IA bots security vulnerabilities.

1.5 Research Objectives and Research Questions

As more business organizations rely on RPA and IA bots to automate processes such as invoice processing as a means to cutting costs and increasing efficiency, the more

critical it is to provide the necessary security measures to ensure that RPA and IA bots do not become soft targets for cyberattacks that exploit the bot's access to client's sensitive personal data. The core research objective of this thesis is to develop a Blockchain-based Intelligent Automation (BCIA) bot to process invoices between business vendors that provide the necessary characteristics to be efficient and secure from data leakage and digital fraud. Also, another research objective is to compare and analyze different blockchain consensus algorithms to identify the one that best supports IA bot operations for secure, efficient, and reliable invoice processing.

Invoice processing was chosen as it is considered one of the primary use cases for Intelligent Automation, based on the scientific literature (Desai et al., 2021; Kerutis & Calneryte, 2022a; Sahu et al., 2020; Upadhyay et al., 2021); as it follows a rule-based structure and process, in addition to applying AI and ML techniques, such as Optical Character Recognition (OCR) to extract invoice data and process the data on the invoice processing application. We approach this objective by introducing cybersecurity detective controls by storing activity logs and invoice payment receipts on BC. The following research questions lay out the pathway to guide the research and achieve its objectives:

RQ1: What are the key security challenges and remedy solution opportunities associated with implementing IA bots powered by Blockchain in B2B e-commerce, particularly invoice processing?

RQ2: How do these challenges and opportunities impact small and medium-sized enterprises' overall efficiency, accuracy, and cost-effectiveness of invoice processing systems?

RQ3: Which BC consensus algorithm provides optimal performance for BCIA bots for invoice processing?

1.6 Research Contribution to the Field

This thesis presents a novel BCIA Bot Framework and application that can help SMEs and large enterprises. The contributions and key features of this thesis can be summarized as follows:

The proposed BCIA Bot Framework integrates blockchain technologies with RPA and AI capabilities to automate invoice processing for businesses of all sizes. Storing activity logs and invoice payment receipts in an immutable BC ledger allows cybersecurity specialists and digital forensics experts to follow the audit trail that is secure and tamper-proof.

There is a significant gap in the literature regarding RPA security in general. This thesis highlights eight critical security challenges facing RPA and IA solutions, thus providing a solid foundation for other researchers to expand their research in this nascent field.

The development of an open-source BCIA bot is the main contribution of this thesis. It can be utilized as a base for further expansions and to implement complex business process automation, for example, for different business needs and scenarios, by researchers and practitioners.

This thesis also addresses which BC algorithm consensus works most efficiently and cost-effectively to develop future BC-based RPA or BCIA bots.

1.7 Research Method:

To accomplish the objectives and answer the research questions posed in this thesis, the Design Science Research Methodology (DSRM) was utilized to develop the BCIA Bot Framework and implement the BCIA bot in a small business simulation utilizing the BCIA bot to automate invoice processing. The initial BCIA Bot was developed using a two-layer architecture: a Cloud layer and a Blockchain layer. The Cloud layer comprises the IA bot, developed in Python, with Optical Character Recognition (OCR) capabilities, which accesses the company's email address and extracts the required invoice data from the invoice files received in PDF format. The BCIA bot then accesses Invoice Ninja, an open-source invoice processing application, to enter the extracted data and process the invoice, followed by the payment processing and, finally, producing a payment receipt. The bot generates an activity log of all the bot's actions with timestamps and machine metadata used to initiate the bot. The Blockchain layer consists of a Smart Contract (SC) that creates and stores a copy of the activity log and evidence of a payment receipt onto the BC ledger.

The primary research method utilized is experimentation based on a real-world scenario of a small business utilizing the BCIA bot to process invoices. The experiment consists of three versions of the BCIA bot based on the BC consensus algorithm and public network used: Ethereum (ETH), Solana Devnet, and BNB Chain. The Ethereum (ETH) BCIA Bot was built in Python and Solidity for the SC, and BNB Chain utilized an Ethereum Virtual Machine (EVM) that works similarly to the ETH BCIA Bot. However, Solana Devnet is built on Rust and requires Solang, a Solidity compiler, to create the transactions. The metrics selected to measure each version's performance were Network throughput, Block size, Transaction fees, Transaction Latency, and Finality Time.

1.8 Thesis Outline

The thesis encompasses five chapters: Chapter One provides a brief background and context of the research, highlighting the research problem, the research objectives and questions, and its significant contributions to cybersecurity and RPA researchers and practitioners in the same field. Chapter Two is a literature review that first consists of a systematic review of the current trends in RPA and IA security challenges, then highlights the gap in the literature regarding utilizing BC technologies in providing security to RPA and IA bots. Next, Chapter Three discusses the methodology and the BCIA Bot Framework in detail. Chapter Four presents the experiments and the subsequent results, then discusses previous literature. Finally, Chapter Five draws the final conclusions and recommendations for the thesis, highlighting the objectives achieved and providing insight for further research directions.

1.9 Summary

In summary, this chapter introduced the topic and context of this thesis: Developing a Blockchain-based Intelligent Automation bot to combat digital fraud and data leakage for B2B e-commerce, precisely automating invoice processing between businesses. It discussed the research objectives and research questions of the thesis and highlighted the significant contributions of the research for both researchers and practitioners. It also presented a brief look into the methodology and structure of the thesis. The developed BCIA Bot Framework provides a security solution that protects and maintains the integrity of the bots' activity logs from tampering or deletion and stores invoice payment receipts on a BC ledger to provide the necessary audit trail to find the culprits of digital fraud. It also deduced that a Proof of Stake

(PoS) consensus algorithm provides the most efficient results in securing BCIA bot activity logs.

Chapter 2 Literature Review

2.1 Overview

This chapter will first provide a comprehensive background on several topics related to the research objectives, including BC components such as blocks, transactions, cryptographic hashing, Smart Contracts, consensus algorithms, and their characteristics, in addition to the role of BC Testnets in developing BC applications. Secondly, following the background section, the chapter will discuss the methodology and results of a systematic literature review on RPA and IA security challenges, in which eight challenges were synthesized from the results. Thirdly, the researcher will present previous literature on BC-based applications, including use cases of BC-based RPAs, to pinpoint the gaps in the literature in which this thesis' research questions will address, as stated earlier in Chapter One.

2.2 Blockchain Technology

Very few technologies today can be labeled as disruptive as BC. Since the publication of Satoshi Nakamoto's whitepaper titled "Bitcoin: A Peer-to-Peer Electronic Cash System" in 2008 (Nakamoto, 2008), software engineers, entrepreneurs, cybersecurity experts, activists, artists, and investors large and small have utilized the technology in a myriad of different applications from cryptocurrencies to Decentralized Apps (DApps) to Non-Fungible Tokens (NFTs) and beyond. Hameed et al. (Hameed et al., 2022) describe BC's evolution over five iterations, starting from the creation of cryptocurrencies (BC 1.0) and then the introduction of Smart contracts (BC 2.0). With the rise of Decentralized apps (DApps) outside the realm of cryptocurrencies, BC 3.0 ventured into Healthcare,

Supply Chain, and Notary systems, amongst others. BC 4.0 shifted the application of BC technologies into heavy industries and manufacturing through the Internet of Things (IoT), Smart Cities, and the energy sector. The researchers believed the current iteration of BC technologies would integrate with AI applications within the Web 3.0 sphere.

BC is, first and foremost, an Internet technology that allows users to create decentralized data structures, utilizing Distributed Ledger Technology (DLT) that records transactions across multiple peer-to-peer computer nodes (Beck et al., 2021). BC's core characteristic is immutability, which means that whatever is recorded on the DLT cannot be changed or tampered with. The DLT also ensures transaction transparency, distributed trust, and consensus between all stakeholders in the chain, as everyone can gain read-only access to the ledger and previous transactions (Chaudhuri et al., 2022).

The components of a typical BC application include blocks, which are the foundational units of data storage in a BC. Each block stores a batch of verified and encrypted transactions, which makes attempts to manipulate or tamper with data futile. Blocks also contain a reference to the previous block, thus creating a chain of blocks (Alexopoulos et al., 2019). Each block typically includes a header, which stores metadata, such as a timestamp, a hash pointer that is a reference to the previous block, and a nonce, a random number used in the mining process. Blocks also have bodies that store the actual transaction data.

Transactions in BC are the records of data exchanges, for example, the exchange of cryptocurrency from one account to another or the transfer of NFT ownership from one entity to another. Transactions also denote the execution of smart contracts or storing data on the block (Filippi et al., 2021). Transactions naturally contain data about the sender, the receiver, the amount transferred, and any additional and

relevant data required to execute the transaction itself. When transactions are initiated within the BC, the network broadcasts the transaction to all nodes to validate the transaction. The process of validating transactions and then adding them to the BC is referred to as consensus.

To secure the integrity of the data stored within the BC, Cryptographic hash functions are utilized to encrypt each block with the hash from the previous block, thus creating a link between all the blocks in the chain. This makes detecting any attempt to tamper with the data very easy, as it would change the hash value and break the chain (Austin & Di Troia, 2022). Public/Private Key Infrastructure (PKI) in BC networks help provide secure authentication, authorization, and encryption. Users generate a pair of cryptographic keys: a public key that is shared openly and serves similarly to an address where others can send messages and transactions. The second private key is not shared publicly and is kept secret. The private key's role is to sign transactions, prove ownership of the transaction, and authorize the transfer of assets over the BC network (Courtois et al., 2014).

Another important component of BC technology is Smart Contracts (SC). SCs are digital contracts that execute and enforce pre-defined rules and conditions when a specific trigger event occurs. The main purpose of smart contracts is to enable trustless and tamper-proof transactions and agreements without the need for middlemen like banks or any third parties (Eggers et al., 2021). SCs on the Ethereum network are written in the Solidity high-level programming language. The SC consists of code that specifies the rules, conditions, and actions needed to be performed. All transactions in SCs follow the same process and algorithm, where the transaction is first created in the contract and then signed by the sender's private key to authorize the transaction for propagation. The transaction is now broadcasted to the Ethereum network and relayed from node to node until it reaches a miner or

validator node, depending on which consensus algorithm is used. The miner or validator validates the transaction and checks if the sender has sufficient funds, or ‘gas’ costs, to cover the transaction value. Next, the valid transactions are finally executed by the Ethereum Virtual Machine (EVM), which updates the state of the Ethereum BC. Each transaction performed by the EVM consumes ‘gas,’ the unit of computational effort required to execute transactions and SCs. Finally, once a transaction is included in a block with other transactions and added to the BC, the transaction is considered finalized. It cannot be circumvented nor modified, as its effects are immutable and permanently recorded on the BC.

Implementing BC in business enterprises is complicated and hard-to-manage components such as P2P interconnection, file storage, smart contract execution need to be developed and tested thoroughly. Blockchain-as-a-Service (BaaS) is a concept derived from cloud computing services such as SaaS or IaaS, while BaaS

2.3 Consensus algorithms

Consensus algorithms play an essential role in ensuring the safety and efficiency of blockchain applications, where selecting an unsuitable algorithm will considerably decrease the application's performance (Mingxiao et al., 2017). A consensus algorithm is a step-by-step process the nodes in a peer-to-peer Blockchain network follow to achieve a joint agreement on the current state of its distributed ledger (“Consensus Algorithms in Blockchain,” 2019). The joint agreement is the only accepted truth in the network. Many consensus algorithms are utilized in BC networks and applications. A list of the most widely used algorithms is as follows:

2.3.1 Proof of Work (PoW)

Famously known as the algorithm used in Satoshi Nakamoto's original Bitcoin whitepaper (Nakamoto, 2008), it relies on the concept of competition between nodes as they calculate solutions to challenging mathematical problems, where the first node to solve the problem gets a reward in the currency of the blockchain network (usually in Bitcoin). PoW's main criticism revolves around the algorithm's high energy expenditure across hundreds or thousands of nodes competing to solve the problem – or mining – to create a new block on the chain.

2.3.2 Proof of Stake (PoS)

Recently adopted by the Ethereum network as an alternative to PoW and its resource wastefulness, PoS reaches consensus through nodes labeled as validators. Validators invest and lock up a quantity of their coins as stakes in the network. The validator for making the next block is selected based on their pool of shares (stakes) in the network semi-randomly (Bamakan et al., 2020). The Follow-the-Satoshi (FTS) algorithm is utilized to execute the stake-based leader selection (Nguyen et al., 2019). The algorithm takes a seed, a string of arbitrary length, as an input, then outputs a token index. The algorithm searches the transaction history using the index to find and select the owner of that token as the leader. When taken into account, as shown in the equation (1) (Nguyen et al., 2019), the probability p_i that node i is chosen to lead a network of N participants is:

$$p_i = \frac{s_i}{\sum_{j=1}^N s_j}$$

(1)

As S_i is participant i 's stake in the network. Therefore, the more stake a node holds, the higher the probability of being selected as leader. Fast block creation time, high throughput, and a 99.5% energy reduction compared to PoW (Ampel, 2023) are some of the advantages of PoS. The main drawback is that having higher-stake nodes sacrifices the blockchain network's decentralized nature.

2.3.3 Delegated Proof of Stake (DPoS)

Mainly targeted for private blockchains, DPoS is a semi-centralized algorithm that allows nodes in the network to select representatives through a voting system to delegate the task of validating blocks. This algorithm's main improvements are better scalability, energy efficiency, and lower-cost transactions.

2.3.4 Proof of Elapsed Time (PoET)

Considered to be the fairest consensus algorithm and widely used in private blockchain networks, PoET gives every validator on the network an equitable chance to create blocks. All nodes wait a randomly generated amount of time and add proof in the block. The created blocks are shared on the network for others' consideration (Guo & Yu, 2022). The validator with the most negligible timer value in the proof is the winner. The fairness of this algorithm stems from the fact that the algorithm provides checks to stop nodes from consistently winning the election and from repeatedly generating the lowest timer value.

2.3.5 Practical Byzantine Fault Tolerance (PBFT)

Originally developed to solve the Byzantine Generals' problem, the BFT consensus algorithm can tolerate having up to 33% of the nodes in a distributed network being faulty or malicious. The main objective of PBFT is to employ collective decision-making in the network to protect itself from any system failures (“Practical Byzantine Fault Tolerance(pBFT),” 2019). Having fewer nodes to reach a consensus means better energy efficiency and higher throughput. On the other hand, PBFT's main drawback is its lack of scalability and possible delays in gathering the needed votes (Bamakan et al., 2020).

2.3.6 Proof of Authority (PoA)

Unlike other algorithms such as PoW and PoS, PoA's main characteristic is the utilization of trusted validators or authorities to validate transactions and add new blocks to the chain. PoA can be used in permissioned and enterprise blockchain networks but has a higher security risk as it relies on a small group of validators that attackers can target (Whiteboard, 2023).

2.3.7 Proof of History (PoH)

Developed by Solana, PoH helps lower transaction costs and facilitates faster transaction processing (Weston, 2023). PoH is a high-frequency verifiable delay function that requires several steps to evaluate yet produces unique outputs that can be verified publicly and efficiently (Yakovenko, 2020).

2.3.8 Proof of Stake Authority (PoSA)

As the name suggests, this consensus algorithm is a hybrid of PoS and PoA algorithms. Its main objective is to address the criticisms of both algorithms by introducing a 1:1 democratic validator model (*Proof of Stake Authority (PoSA) Definition | CoinMarketCap, 2023*) where each validator has an equal stake in the network. PoSA gained popularity as the algorithm used in Binance's BNB Chain network. However, PoSA sacrifices decentralization and security as a trade-off for short block times and lower costs (Shrivastava, 2022).

2.3.9 Proof of Capacity (PoC)

Unlike PoW or PoS, which depend on computational power or stake ownership, respectively, PoC leverages the hard disk storage capacity as the basis for block validation. Participants in a PoC system are known as farmers, and each farmer will allocate storage space on their devices to store cryptographic hashes or plots. Plots contain pre-computed solutions to cryptographic puzzles and challenges (Tang et al., 2021). When new blocks need to be created on the BC, farmers select portions of their storage space and present it as proof of their capacity to participate in block validation. The farmer who finds a valid solution to the puzzle first is rewarded with the right to create and append the new block on the BC and the transaction fee (Kumzy, 2024). PoC is considered to be more than 30 times more energy-efficient in comparison to PoW and can mine blocks in half the time (four minutes vis-à-vis ten minutes on a PoW network) (*Proof of Capacity (PoC) in Blockchain - Shiksha Online, 2023*).

2.3.10 Proof of Importance (PoI)

PoI is designed to encourage and incentivize participation and contribution to building new blocks on the BC by rewarding users based on their Importance score (“Proof of Importance,” 2023). The Importance score is based on three main factors: Transaction Volume, where the higher the volume of transactions from a particular user, the more critical they are to the network. Account Balance is another main factor, where users with more significant balances of the native cryptocurrency are given higher importance. While Account Balance plays a significant role in gaining importance, developing PoI was a response to combat coin hoarding in PoS networks, which gave them an unfair advantage, where the rich keep getting richer by gaining all the rewards and incentives in the form of gas transaction fees, whilst other participants do not get the opportunity to benefit from the transaction fees due to having fewer coins (Ferdous et al., 2020). Finally, the user’s Reputation increases their importance in the network, which is measured by the trustworthiness and reliability of a user (Xiao et al., 2022).

2.3.11 Raft

Originally developed as a simpler alternative to the more complex Paxos algorithm (Huang et al., 2020), The Raft CA is designed to achieve consensus among a set of nodes in a distributed system. Raft is widely used to ensure fault tolerance and consistency. In Raft, the nodes are organized into clusters, with one node elected as the leader in each cluster. The leader’s role is to coordinate the replication of log entries across the cluster. Raft achieves consensus by replicating a log of commands or entries across the cluster. The leader is responsible for appending new entries to the log and ensuring all followers replicate the entries. New leaders are elected when

the original leader becomes unavailable for whatever reason(Ongaro & Ousterhout, 2014).

2.3.12 Comparison of Consensus Algorithms

Several publications in the literature have attempted to categorize and compare the performance and utility of consensus algorithms (CA). Hattab and Alyaseen’s work (Hattab & Taha Alyaseen, 2019) classified the CAs into three main groups: Proof based on hardware, which denotes CAs that rely on hardware to perform computational tasks such as PoW, PoET, and PoC. The second group revolves around stake-based consensus, which favors nodes' overall stake in the network rather than their energy consumption. This group consists of PoS, PoI, and Leased PoS. The third group are CAs that use proof based on voting and comprise PBFT and other variations such as DBFT and FBA.

(Mingxiao et al., 2017)’s research drew a comparison of five consensus algorithms based on five characteristics, as shown in Table 1. Byzantine fault tolerance is a property of BC networks that enables them to continue functioning correctly and reaching consensus even when experiencing some nodes acting maliciously or arbitrarily, otherwise known as Byzantine faults (“Practical Byzantine Fault Tolerance(pBFT),” 2019). On the other hand, crash fault tolerance denotes the resiliency of a CA when faced with a crash failure, where a node in the system stops abruptly (Podgorelec et al., 2019). The verification speed refers to the time it takes for a transaction or block to be verified and confirmed by the BC network participants. It is affected by numerous factors, including network congestion, block size, and transaction complexity. Throughput, typically expressed as the number of transactions processed per second (TPS), is an essential metric for measuring the rate BC processes and confirms transactions within a given timeframe (Huang et al.,

2020). Scalability denotes the ability of a BC network to handle increasing transaction volumes, users, and data without significant loss of performance, efficiency, or security. According to Table 1, despite having higher throughputs than PoW, PoS, and DPoS, both PBFT and Raft CAs have weak scalability.

Table 1 Comparison of five Consensus Algorithms (Mingxiou et al.,2017)

Characteristics	Consensus algorithms				
	PoW	PoS	DPoS	PBFT	Raft
Byzantine fault tolerance	50%	50%	50%	33%	N/A
Crash fault tolerance	50%	50%	50%	33%	50%
Verification speed	>100s	<100s	<100s	<10s	<10s
Throughput (TPS)	<100	<1000	<1000	<2000	>10K
Scalability	Strong	Strong	Strong	Weak	Weak

A different approach to categorizing CAs was developed by (Ferdous et al., 2020), who argue that CAs can be categorized as either incentivized or non-incentivized. Incentivized consensus is the CAs that reward participants, usually in transaction fees, for creating and adding new blocks in the BC. Incentivized consensus is found predominately in public BC networks, as the rewards provided ensure that all participating nodes follow the corresponding consensus protocol at all times. However, Non-incentivized CAs are found exclusively on private blockchain networks, where all nodes are previously authorized and trusted; therefore, there is no need for a reward system to be in place. Another significant contribution of their work is the development of a decision tree to help blockchain developers decide which CAs are most suitable for their DApps based on a multitude of criteria,

including incentivized or non-incentivized, high or low energy consumption, scalability, security, and Application-Specific Integrated Circuit (ASIC) resistance. ASIC-resistant blockchains and mining algorithms are designed to disincentivize users who own specialized hardware to mine cryptocurrency by giving them no advantage over users using conventional Graphical Processing Units (GPU) mining (*ASIC-Resistant*, 2023).

2.4 Testnets

Testnets are identical copies of an existing blockchain network to create a sandbox environment for developers to test out new BC products or features. Faucet websites provide BC developers the necessary tokens or currency, usually referred as “gas,” to be able to run their application on the blockchain testnet. Unlike the mainnet blockchains, testnet tokens do not have any monetary value. Yet, faucet websites ration minimal tokens daily to avoid token abuse and misuse, and ensure the longevity of the testnet.

Table 2 displays the currently available Blockchain Testnets to test DApps and their consensus algorithms. Ropsten and Goerli are two testnets that were deprecated in 2023 and January 2024, respectively. Ropsten's deprecation results from Ethereum's 2.0 launch and migration from a PoW network to a PoS network. Goerli's deprecation came after facing token supply issues and heightened demand for test ETH, which led to its unintentional gaining of monetary value, thus defeating the purpose of the testnet. Sepolia is the current replacement for Goerli testnet, as it uses the same PoS algorithm and is supported by Ethereum.

Table 2 Blockchain Testnets and their corresponding consensus algorithms

No.	Testnet	Consensus Algorithm	Availability
1	Goerli	Proof of Authority	Deprecated Jan 2024
2	Sepolia	Proof of Stake	Available
3	Ropsten	Proof of Work	Deprecated 2023
4	Holesky	Proof of Stake	Available
5	Solana Devnet	Proof of History	Available
6	Mumbai	Proof of Stake	Available
7	BNB Chain	Proof of Staked Authority	Available

2.5 Security Challenges in RPA and IA

Robotic Process Automation (RPA) is an increasingly disruptive technology that automates business processes through simple conditional if-then-else logical rules while interacting with numerous Information systems (IS), primarily through Graphical User Interfaces (GUI) screen recordings (Ivančić et al., 2019). RPAs are a product of the 4th Industrial Revolution, where, historically, physical robots replaced blue-collar jobs, and software robots will possibly replace white-collar jobs within the next few years (Madakam et al., 2019).

The evolution of RPAs, namely cognitive RPA and Intelligent Automation (IA), combines AI to enhance the RPA's capabilities to handle business process automation (Ng et al., 2021). IA took process automation from a structured, rule-based, and rigid standpoint towards providing automation solutions that can analyze and process unstructured data and utilize AI technologies such as computer vision, natural language processing, and fuzzy logic decisions (*Becoming Strategic with Intelligent Automation*, 2020).

Organizations with poor RPA security management and implementation can negatively affect their business. RPAs are vulnerable to interrupted or sudden process stoppages, significantly impacting business continuity (Giesbers, 2020).

Authorizing bots to access private data unchecked can lead to serious compliance violations, and companies can expect authorities to hand down heavy fines and penalties (Murugappan & Sree Kala, 2022).

Utilizing IA for bespoke tasks and cases with unexpected or unused factors can be a hindrance and a significant security risk to the enterprise (Kaplan et al., 2019; Smith, 2020). Andriole questioned what would happen when AI is fully integrated within RPA bots and believed that it is impossible to conclude that the rise in digital maturity will somehow reduce digital risk (Andriole, 2020).

The current scientific literature does not explore the many security challenges in developing RPA or IA bots to combat cybercrimes such as data leakage and digital fraud, nor does it address RPA or IA security challenges from a technical point-of-view. As shown in Table 3, the researcher compared six leading publications on RPA security challenges in the five-year period from 2019 to 2023. Five of the six publications discussed the managerial or business-oriented perspective of RPA security. In contrast, only one publication (Kosi, 2019), tackles the technical perspective exclusively. One publication from 2020 (Enriquez et al., 2020) and 2023 (Kurylets & Goranin, 2023) discuss the technical aspects briefly, which gives the reader a clear indication that scientific literature regarding RPA and IA security risks and how to tackle them from a technical approach is severely lacking up till 2023.

Table 3 RPA security challenges and their categories according to Previous Literature

Publication Title	Challenges	Categories
Robotic Process Automation (RPA) and Security [Thesis] (Kosi, 2019)	• Abuse of Privileged Access • Data Leakage • Security Vulnerabilities • Denial of Service	Technical
An Enhanced Security Framework for Robotic Process Automation [Journal] (Murugappan & Sree Kala, 2022)	• Governance • Risk Management and indicators • Access Control management • Incident and continuity management • Software Security • Compliance and Auditing management	Managerial/ Business oriented
Robotic Process Automation: Contemporary Themes and Challenges [Journal] (Syed et al., 2020)	• Benefit Metrics and realization support • infrastructure and organizational assessments • Adoption and implementation support • Task selection and analytical capabilities • Critical Success factors • Proactive monitoring and exception handling	Business-oriented
Robotic Process Automation: A Scientific and Industrial Systematic Mapping Study	• AI Implementation • Lack of Analysis in RPA development • Lack of Testing	Technical/ Managerial/ Social

[Journal] (Enriquez et al., 2020)	• RPA Impact on employee competence and well-being	
Robotic Process Automation Risk Management: Points to consider [Journal] (Hong et al., 2023)	• Risk Management	Managerial
Security Ontology OntoSecRPA for Robotic Process Automation [Journal] (Kurylets & Goranin, 2023)	• Data Leakage • Risk Management • Cyberattacks	Managerial/ Technical

This section contributes to the scientific literature by compiling the current security challenges and provides a much-needed foundation for future RPA and IA security research. This section presents a Systematic Literature Review (SLR) on the current state of RPA and IA and their security challenges. This review utilized five databases: Scopus, IEEE Explore, ACM Digital Library, and ScienceDirect. The SLR will primarily address the following three research questions:

SLR-RQ1: What is the current state-of-the-art regarding Robotic Process Automation (RPA) and Intelligent Automation (IA)?

SLR-RQ2: According to the currently available literature, what are the cybersecurity challenges in RPA and IA?

SLR-RQ3: Based on the current literature, can Blockchain technologies provide viable and feasible cybersecurity solutions for RPA and IA?

2.5.1 SLR METHODOLOGY

This systematic literature review utilizes the Kitchenbaum method and its updated method by Petersen et al. (Kitchenbaum & Charters, 2007; Petersen et al., 2015).

The initial search yielded 152 publications from five databases:

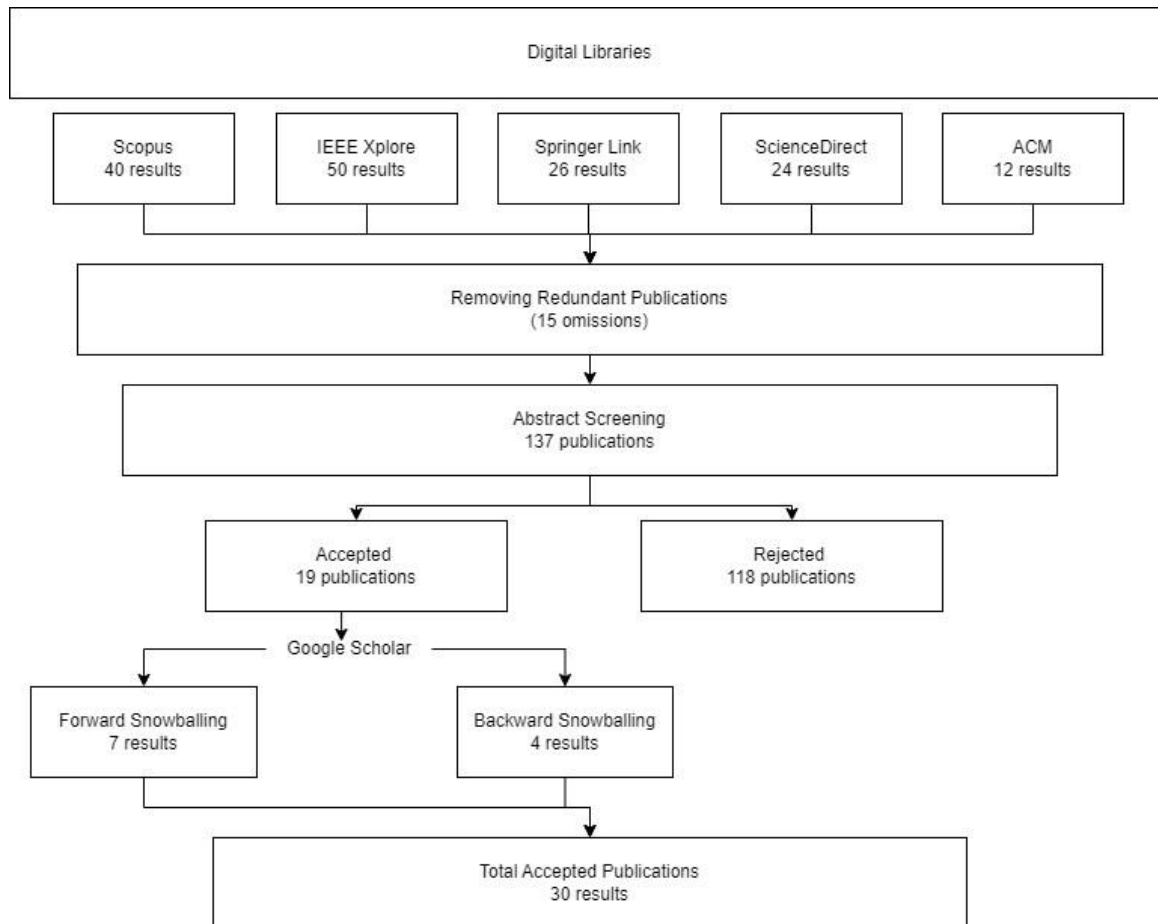


Figure 1 Initial SLR and Snowballing Process

Scopus, IEEE Xplore, Springer Link, Science Direct, and ACM Digital Library. Table 4 displays the search terms and filters used in the initial publication search. The search terms included Robotic Process Automation, Intelligent Automation, and their abbreviations, in addition to the terms “security”, “risks”, and “fraud” with

some variations depending on the database. As for the subject areas, the search was confined to the fields of computer science, information technology, and business studies. The search period was five years, from Jan 2017 to May 2022.

Table 4 Initial Search Terms by Digital Library

Digital Library	Query
Scopus	TITLE-ABS-KEY ("Robotic Process Automation" OR rpa, AND automation, AND cybersecurity OR security) A ND (LIMIT-TO (SUBJAREA , "COMP") OR LIMIT-TO (SUBJAREA , "ENGI") OR LIMIT-TO (SUBJAREA , "BUSI")) (40 results)
IEEE Xplore	("Abstract": "Robotic Process Automation", RPA) OR ("Abstract": "Intelligent Automation") AND ("All Metadata": Security) Refined By: Publication Topics: business data processing, learning (artificial intelligence), software agents, artificial intelligence, data mining, user interfaces, organizational aspects, security of data, fraud, human-computer interaction, human-robot interaction, insurance data processing Year: 2017-2022 (50 results)
Springer Link	"Robotic Process Automation" AND (Security, OR Risk*, OR fraud), Discipline: Computer Science, Subdiscipline: IT in Business, Systems and Data Security, Artificial Intelligence, Computer Applications, Computer Science. Year: 2017-2022 (26 results)
ScienceDirect	"Robotic Process Automation" OR "Intelligent Automation", AND Security, Fraud, Risk (24 results)
ACM	[[Abstract: "robotic process automation"] OR [Abstract: "intelligent automation"]] AND [[All: security] OR [All: fraud] OR [All: risk*]] AND [Publication Date: (01/01/2017 TO 31/05/2022)] (12)

2.5.2 Inclusion/ exclusion criteria

Terms such as Robotics and the Internet of Things (IoT) were excluded from the initial search as RPAs and IA bots are software robots and do not require specific hardware components. The search results were checked and screened for redundancies across the databases, which led to the omission of 15 research articles, leaving the total results to 137 publications. The second screening of the publications' abstracts led to the acceptance of 19 publications and the removal of 118 publications, as shown in Figure 1.

2.5.3 Snowballing

The final screening round was applying backward and forward snowballing to the 19 remaining publications. Snowballing, according to Wohlin (Wohlin, 2014), refers to utilizing a publication's reference list or citations to identify additional publications for systematic literature reviews as a means to find more relevant results to serve the SLR . A snowballed publication's relevance is determined by where the paper is referenced within the original published text or where the paper is cited in other academic publications. Google Scholar, a specialized search engine for academic publications, conveniently displayed the links to the citations of each paper and was thus used as the search tool for the forward snowballing step.

The snowballing procedures led to the addition of 11 publications: seven publications from forward snowballing, including three postgraduate theses and four journal articles, and four publications from backward snowballing comprised two conference proceedings, a journal article, and a postgraduate thesis, thus raising the total to 30 publications as shown in Figure 1. The definitive collection of

publications comprised 17 conference proceedings, nine journal articles, and four postgraduate theses (Masters, PhD), as shown in Table 5.

Table 5 Frequency of SLR-selected publications

Publication Type	Publications	Frequency
Conference Proceedings	(Basile et al., 2021) , (Bhamidipati et al., 2021) , (Biswas & Dutta, 2021) , (Chakraborti et al., 2020) , (Chen et al., 2022) , (Choi et al., 2021) , (Griffiths & Pretorius, 2021) ,(Iqbal & Matulevičius, 2019) , (Kurian & Varghese, 2021) , (Money, 2021), (Müller et al., 2020), (Murugappan & Sree Kala, 2022), (Ravi & Kamaruddin, 2017), (Raza et al., 2019), (Suri et al., 2017), (Syed & Wynn, 2020), (Thekkethil et al., 2021)	17
Journal Articles	(Dahlia Fernandez & Aini Aman, 2021) , (Flechsigt et al., 2022) ,(Gotthardt et al., 2020) , (Costa et al., 2022) , (Marciniak & Stanisławski, 2021) , (Moffitt et al., 2018) , (Syed et al., 2020) , (Wojciechowska-Filipek, 2019b) , (Zemánková, 2019)	9
Postgraduate theses	(Kosi, 2019),(Lok, 2021), (Man, 2022) , (Noppen & Jonker, 2019)	4
	Total	30

Across the period from January 2017 to May 2022, 2021 recorded the highest number of publications, with eight conference proceedings, three journal articles, and one postgraduate thesis, as presented in Figure 2. One can speculate that this spike in interest in RPA and IA security stemmed from increased interest in Automation due to the COVID-19 pandemic. The first half of 2022 had a similar

output to 2019, which may indicate a growing interest in RPA and IA security despite an initial dip from 2021.

At first impression, this result shed light on the wide disparity in interest and research between practitioners and academia regarding RPA and IA bot security solutions due to the relatively low number of publications in this particular field as discovered in this SLR.

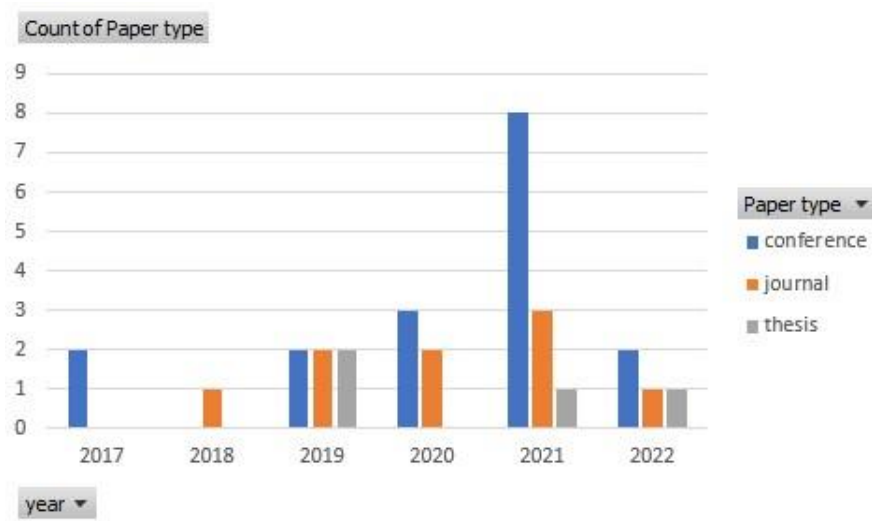


Figure 2 Number of Publication types by year (Jan 2017- May 2022)

2.6 Results

According to the literature, eight types of security challenges were synthesized from the final selection of publications to answer SLR-RQ2 regarding the cybersecurity challenges facing RPA and IA. The following challenges are listed by topic frequency within the publications: Logging and Auditing, Access Control, Security Standards and Compliance, Encryption, Blockchain Security vulnerabilities, Network Security, AI security vulnerabilities, and Lack of Testing, as shown in Table 6.

Table 6 RPA and IA Security Challenges based on SLR

No.	Publication	Security Issues							
		Logging and Auditing	Access Control	Security Standards and Compliance	Blockchain Security vulnerabilities	Encryption	Network Security	AI security vulnerabilities	Lack of Testing
1	(Chakraborti et al., 2020)	✓							
2	(Griffiths & Pretorius, 2021)								
3	(Suri et al., 2017)		✓						✓
4	(Ravi & Kamaruddin, 2017)	✓							
5	(Syed & Wynn, 2020)		✓						
6	(Basile et al., 2021)				✓				
7	(Müller et al., 2020)				✓				
8	(Murugappan & Sree Kala, 2022b)	✓	✓						
9	(Dahlia Fernandez & Aini Aman, 2021)		✓				✓		
10	(Kurian & Varghese, 2021)		✓			✓	✓		
11	(Iqbal & Matulevičius, 2019c)							✓	

12	(Biswas & Dutta, 2021)							✓	
13	(Choi et al., 2021)								
14	(Raza et al., 2019)	✓	✓			✓	✓	✓	
15	(Wojciechowska-Filipek, 2019)		✓						
16	(Syed et al., 2020a)	✓	✓	✓			✓		
17	(Bhamidipati et al., 2021)				✓				
18	(Thekkethil et al., 2021)	✓							
19	(Kosi, 2019)	✓	✓	✓		✓			✓
20	(Man, 2022.)	✓	✓			✓		✓	✓
21	(Noppen & Jonker, 2019)	✓	✓	✓					
22	(Gotthardt et al., 2020)	✓	✓	✓					
23	(Costa et al., 2022)			✓					
24	(Marciniak & Stanisławski, 2021)		✓	✓					
25	(Flechsigt et al., 2021)	✓	✓	✓					
26	(Lok, 2021)	✓	✓	✓					
27	(Zemánková, 2019)	✓			✓			✓	
28	(Moffitt et al., 2018)	✓							
29	(Money, 2021)	✓							
30	(Chen et al., 2022)						✓		
	Frequencies	16	15	8	4	4	4	4	3

2.6.1 Logging and Auditing

Researchers have a wide consensus on the requirement of robust logging and auditing capacities in RPA bots to locate and mitigate technical issues and provide digital forensic evidence to uncover any data tampering or leakage from malicious actors. This issue lands at the top of the list regarding data leakage and digital fraud, with 53% of the selected publications highlighting this issue, as shown in Table 6.

The terms logging and auditing are used interchangeably. However, each serves a specific purpose. Logging, on the one hand, records every action from a user or entity in a system to monitor who acted and when. On the other hand, auditing or audit trails rely on the sequence of actions recorded to answer how and why the system is in a particular state.

Murugappan and Sree's enhanced security framework for RPAs considered audit and compliance as one of the significant pillars of their framework (Murugappan & Sree Kala, 2022). Financial service companies heavily rely on providing evidence and audit trails of all processes and transactions to comply with security and privacy laws and regulations (Chakraborti et al., 2020; Ravi & Kamaruddin, 2017). In addition, audit trails provide the necessary step-by-step forensic analysis to pinpoint and understand RPA bot process failures. As more RPA bots are deployed within an IS, the importance of logging and auditing increases exponentially (Lok, 2021).

Despite the universal consensus on the necessity of logging and auditing, the literature lacks robust proposals or solutions to provide complete, accurate, and trustworthy records. Thekkethil et al. briefly addressed this by stating that BC technology is essential in auditing and protecting data privacy, especially for transactions and smart contracts (Thekkethil et al., 2021).

2.6.2 Access Control

Access Control in the RPA context denotes a bot's access privilege to sensitive data within an IS, in addition to the bot's login credentials and whether human employees can access the bots' accounts legitimately or with malicious intent. According to Table 6, Access control was discussed in half of the selected publications. Researchers shared opposing opinions regarding the risks of unauthorized data access by RPA bots.

For example, (Suri et al., 2017) believed that RPA bots had no risk of unauthorized data access and could not modify nor enhance the clients' applications. An RPA developer interviewed in Syed and Wynn's paper titled *How to Trust a Bot: An RPA User Perspective* quoted that, *"From a security perspective, bots had their own logins so that there was no sharing in that respect, so that did not become an issue."* (Syed & Wynn, 2020). To secure bots from password tampering, some organizations authorize bots to change their own passwords, which an authorized human employee can retrieve when required (Noppen & Jonker, 2019). In addition, according to (Wojciechowska-Filipek, 2019a), using RPA bots was a better alternative to human employees in regards to protecting and processing confidential information such as passwords (Al-Slais & El-Medany, 2022).

However, the opposing view sees that bots' user credentials cannot be secured adequately (Flechsigt et al., 2022). Companies might postpone or neglect basic security measures such as sharing human credentials with RPA bots and insulating duties among different RPA bots (Lok, 2021).

2.6.3 Security Standards and Compliance

Organizations targeting to reduce data leakage and digital fraud must guarantee that the integrity of the data required to complete the business processes is accurate. Data accuracy and reliability can only be achieved by implementing the necessary security controls (Griffiths & Pretorius, 2021).

Security Standards and Compliance are essentially an organization's products of Security Governance. The results of robust and thorough governance fill in the possible flaws and gaps regarding RPA and IA security. However, according to the literature, there is no clear or standard framework for achieving high compliance with RPA and IA development.

2.6.4 Network Security

Providing secure networks for organizations through the implementation of security policies and hardware/ software firewalls is the major preventative measure to limit remotely unauthorized access to RPA and IA bots.

2.6.5 Blockchain Security Vulnerabilities

As for BC technologies, several publications have highlighted the benefits of their decentralized nature and strong encryption, but some security concerns remain as BC technologies are vulnerable to attacks such as Sybil attacks, Database injection attacks, timestamp manipulation, and fraudulent data entry (Bhamidipati et al., 2021).

2.6.6 AI Security Vulnerabilities

Other 4IR elements, such as AI, also had some security vulnerabilities that needed to be addressed. AI concerns include the utilization of black-box models, where

malicious data scientists and developers can potentially create malicious backdoors to manipulate ML models deployed in production (Man, 2022).

2.6.7 Encryption

Researchers pointed out several key areas and data points that require encryption, including bot login credentials, clients' sensitive data stored in databases, and end-to-end encryption through transmission channels.

2.6.8 Lack of Testing

A lack of robust testing has long been a pain point in the Agile software development cycle approach. However, researchers such as Kurian and Varghese warned that a lack of robust and thorough testing could financially lead to catastrophic financial consequences for organizations and relegate RPA and IA bots inoperable (Kurian & Varghese, 2021). Also, a lack of penetration testing leaves RPA bots vulnerable and exposed to the simplest security exploitation.

2.7 RPA and IA for Invoice Processing

Invoice processing is a cornerstone activity in every business. Still, it is notoriously a time-consuming one. Aberdeen Group reported that invoices can take four to sixteen days to be processed (Sahu et al., 2020). As a technological approach to improve invoice processing, Robotic Process Automation (RPA) and its Artificial Intelligence (AI) based iteration, Intelligent Automation (IA), is a fast-growing trend in businesses worldwide. According to Gartner, RPA has grown into a multibillion-dollar industry in 2022 (Carter, 2023)

Utilizing RPA in invoice processing has several benefits, such as being cost-effective in terms of time and money. For example, In Touzon's thesis titled 'Digital

Transformation In Companies: Invoice Processing As A Document Manager’, he found that a large enterprise can handle upwards of 500 invoices per month, whereas if they were handled manually with data entry from paper invoices and manual approvals, each invoice would cost over \$11 (USD) to complete. In contrast, using RPA software to process the invoices, each invoice would cost only under \$1 (USD). The researcher also believed that, on average, small companies that process 50 invoices per month can benefit from RPA software, where savings can reach up to 40%(Touzon & Garuz, 2020). In another survey targeting American companies whose annual revenues exceed \$USD 50 million a year, their organizations on average process 22,039 monthly invoices (*Why Automation Matters: A Survey Study of the Modern Accounts Payable Department*, 2016). According to Levvel Research (*How Much Does It Really Cost to Process an Invoice?*, 2022), the average cost of processing a single invoice manually costs approximately \$USD 19, and when invoice processing is automated, the cost comes down dramatically to \$USD 7 per invoice.

PwC estimated that Automation can save up to two trillion dollars annually worldwide (Gotthardt et al., 2020). Another significant benefit is providing high accuracy and enhanced performance (Sahu et al., 2020). Kerutis and Calneryte noted that human performance decreased as the number of documents processed increased. In contrast, RPA bots performed 1.76 times better than humans in processing 500 invoices and twice as efficiently when handling a thousand invoices (Kerutis & Calneryte, 2022b).

Cybersecurity risks are part and parcel of any emerging technology, and IA is no different. According to CyberArk research (Security, 2022), cybersecurity concerns are a major obstacle in businesses' adoption of RPA and IA solutions, where approximately 25% of security professionals find RPA/IA bots a major identity

security risk in their enterprises. Cybercriminals can utilize malicious software to access IA bots to reveal customers' or users' sensitive data (*Robotic Process Automation Security and Why It's Important*, 2022). Data leakage can also occur as a result of poor RPA bot training or done with malicious intent (Nallasivam, 2022). Another cybersecurity risk is the rise of digital fraud, where malicious actors can defraud businesses by injecting false invoice information and generating money illegally. Unfortunately, introducing security measures such as Multifactor Authentication (MFA) and encrypting bots' actions are not possible in RPA or IA bots (*Reducing RPA Security Risk | Security Magazine*, 2023).

Logs provide information about regular events and raise flags on possible malicious behavior or attacks such as Denial of Service (DoS) attacks, malware activities, and other attacks on an organization's infrastructure (Putz et al., 2019). A widely used approach to cover malicious actors' tracks when compromising a system is through manipulating and tampering with activity logs (Shekhtman & Waisbard, 2021). Therefore, having an accurate and immutable audit trail is of great value to organizations (Sanchez et al., 2022). As a preventative and detective approach to securing IA bots, which inherently has growing cybersecurity concerns, this underlines the urgency to develop a Blockchain-based intelligent automation bot for invoice processing that ensures a robust and immutable activity log to minimize digital fraud and data leakage in IA bots.

2.8 RPA, IA, and Blockchain Use Cases

(Guarda et al., 2024) in their publication titled “Blockchain and Robotic Process Automation Working Together”, they argued that the joint use of BC and RPA can create robust solutions that simplify processes, increase efficiency, and reduce costs. They present examples of sectors that can benefit from utilizing BC and RPA

together, such as in healthcare, to streamline administrative processes and protect patients' data. Another example is in government, where public servants can improve the efficiency of their services through RPAs and reduce the risks of data breaches to sensitive government documents and citizens' data. The researchers also raise concerns about the main challenges of integrating BC and RPAs: technical complexity, data privacy concerns (Al-Slais, 2020), regulatory compliance, scalability, and finally integration with legacy systems.

In the publication titled “Process Automation and Blockchain in Intelligence and Investigation Units: An approach” (Sobreira Leite et al., 2020), the researchers surveyed Brazilian intelligence specialists and law enforcement officers on the feasibility of utilizing process automation and applying BC technologies to protect digital assets and evidence from being tampered during investigations. After a positive initial response, an experimental BC-based process automation application was deployed and led to increased completed evidence requests per month, starting from two to three requests per month to 16 per month after the experiment. It also led to a dramatic decrease in the average time spent per request from 15-20 days before the experiment to only three to eight minutes per request.

RPA and BC have been used to ensure fairness in highly competitive situations. (Sibanyoni, 2021)'s work proposed a system that allowed parents to register their children for their school of choice. Many parents have mistrust issues due to historical precedences of unfair and unjust allocation of educational opportunities at some of the most highly sought-after schools. To regain trust, the researcher proposed a BC-based RPA application that extracts registration form data from parents and timestamps the registration receipt onto a BC to maintain fairness within a ‘first come, first serve’ concept.

BC technologies have been integrated with RPA in financial markets, particularly supporting derivatives (Syed & Wynn, 2020). In other fields, such as supply chain businesses, RPAs can automatically capture data from an enterprise resource planning (ERP) system and add it to the BC (Hartley & Sawaya, 2019). This approach improves the transparency of transactions throughout the supply chain and minimizes the risks of data manipulation and digital fraud risks.

2.9 Blockchain-based Invoice Processing

Several researchers in the literature have discussed how BC-based invoice processing can benefit business enterprises in multiple ways. (Narayanam et al., 2020)'s work developed a BC-based e-invoicing system that enabled efficient invoice dispute settlement between shippers on the high seas and working in global trade. The researchers developed the e-invoicing system utilizing the Hyperledger Fabric platform and a cloud microservices architecture. After conducting several experiments, the researchers found that deploying the system in a real-world situation would be practical. Xidong (Liu, 2018) proposed an alliance chain that gathers government taxation bureaus, invoicing companies, businesses, and individuals under one BC application to manage the whole lifecycle of electronic invoice assets by utilizing smart contracts to execute the business processes. The researcher believed such a system would profoundly impact current fiscal and taxation systems.

Many businesses experience times during the year when there is minimal cash flow and, therefore resort to lending from banks and financial institutions and using their invoices as collateral to obtain funding. (Guerar et al., 2020) developed a fraud-resilient BC-based solution to aid invoice financing and factoring between SMEs and financial institutions. The solution was built on a public BC that supports both

fully open and group-restricted auctioning of invoices. Due to the trustless nature of public permissionless BCs, the researchers introduced a reputation system based on the past history of nodes in the network. By having a reputation system in place, insurance companies can moderate the cost of insurance based on the users' reputation. The researchers believed their system could reduce insurance costs and limit digital fraud. The work by (Battaiola et al., 2019) opted to use a permissioned BC in their invoice factoring solution. The system's main aim is to minimize fraudulent double pledging of invoices to multiple factors. The BC allowed factors to check whether invoices had been already factored (double pledged) without forcing all stakeholders i.e. the banks, factors, and businesses, to share and disclose their data.

2.10 Blockchain for Digital Forensics evidence creation and storage:

2.10.1 Block-DEF

Block-DEF is a BC secure digital evidence framework with loose coupling structures. Block-DEF's structure stores and maintains the original evidence file and the evidence information in separate locations (Tian et al., 2019). The Block-DEF framework is built in a three-layer architecture: Service layer, BC layer, and a Network layer. The service layer focused on supporting evidence-related services by using the BC. It employed a multiple-signature scheme to ensure integrity. The service layer delivers three main processes: evidence creation, submission, and evidence recall. There are three types of beneficiaries in Block-DEF: Content providers who own the original files, whether from a website, or as email messages, or files stored on a file server. Evidence providers collect artefacts to prove file tampering has occurred, while Evidence requesters are the analysts who request evidences provided by the evidence providers.

The BC layer in the Block-DEF architecture is on either a private BC or a consortium block. The researchers reasoned that the private BC network environment is much safer, block mining is non-incentivized, and thus, it is more cost-effective. However, blockchain bloat is the main pain point of having it on a private BC network. Blockchain bloat occurs when the size of a BC grows exponentially over time, mainly due to the increased volume of transactions, block size limitations, and BC's immutable nature as outdated and redundant data remain permanently on the BC (Alzoubi & Mishra, 2024). Block-DEF's assigned CA differs between private and consortium utilizing PBFT or Raft.

The third and final layer, the network layer, provides the communication channel between the service and BC layers. Block-DEF follows the BC convention of utilizing P2P networks to send transactions, encrypt evidence information, and seek consensus among the BC network nodes.

2.10.2 BCFL Logging Framework

(Awuson-David et al., 2021) 's work describes the difficulty of obtaining digital evidence in a cloud architecture where the multi-tenancy, geolocation, and Service-level agreements have added an extra layer of complexity in acquiring digital log evidence from a cloud architecture. Due to this problem, the researchers designed a Blockchain Cloud Forensic Logging (BCFL) framework. The framework's workflow consists of four steps: First, it processes transaction logs using BC's DLT technology. Second, a BC smart contract should be utilized to maintain the integrity of the logs. Third, all the transaction logs must be validated, and finally, the transaction log must be immutable. Utilizing cloud computing makes it unclear where data is located, in addition to the growing need for regulatory compliance with regional data security regulations such as the EU's General Data Protection

Regulation (GDPR) (Al-Slais, 2020). Meanwhile, BC provides auditable and transparent transaction logs, which makes settling legal disputes easier and more efficient. The BCFL framework utilizes Hyperledger Fabric as its BC platform, which supports the concept of channels. Channels are a separate BC that handles secret transactions; therefore, each Fabric client can use a different and unique channel. This approach allows forensic investigators to track and locate evidence more quickly as it circumvents the multi-tenancy issue in cloud systems. The BCFL Infrastructure-as-a-Service (IaaS) cloud ecosystem can interact with other service ecosystems through a REST gateway and authenticated BCFL users.

2.11 Summary

To summarize this chapter, the researcher presented a broad background regarding BC and its technologies, consensus algorithms, and highlighted the differences between the CAs according to the scientific literature. In addition, a systematic literature review of 30 publications was conducted and as a result, synthesized and highlighted eight security challenges regarding RPA and IA bots, with a strong emphasis on the importance of more robust logging, auditing, and access control (Al-Slais & Ali, 2023). The main limitation of this work is the lack of technical publications regarding RPA and IA security challenges. Nevertheless, the SLR results provide a robust and comprehensive foundation for researchers to continue RPA and IA security research from a technical point of view. As for practitioners, these results can guide the development of effective RPA and IA security policies to combat any new security threats in the future.

Moreover, cybersecurity officers can target and develop technical solutions to one or multiple security challenges discussed in this SLR to improve RPA and IA bots' security and minimize the impacts of digital fraud and data leakage. This chapter

also presented some examples of BC-based RPA use cases on one side and BC-based invoice processing on the other side to highlight that there is no overlap in the literature in discussing a BC-based intelligent automation bot for invoice processing. This once again underlines the novelty and contribution of this research.

Chapter 3 Methodology

3.1 Overview

This chapter presents the research methodology utilized in this thesis, Design Science Research Methodology (DSRM) by first illustrating two scenarios and explaining how malicious actors can exploit IA invoice processing bots to commit data leakage and digital fraud. Based on the two scenarios, the researcher proposes the BCIA Bot Framework and its components to implement the BCIA bot to address these security threats. Next, the quantitative method and utilized experiments will be described in detail.

3.2 Design Science Research Methodology

Research Methodology is the systematic process of planning, conducting and analyzing research. Once settled on the research methodology, researchers can outline the techniques, procedures, and strategies needed to investigate a specific topic, answer research questions, or test hypotheses (Pandey & Pandey, 2015). Research methodologies serve as the compass that guides researchers through the various stages of their research journey and ensure that their work is thorough, valid, and adds meaningful contribution (Thomas, 2021).

Initially developed by (Peppers et al., 2007), the Design Science Research Methodology (DSRM) has grown to become a widely accepted methodology and approach in conducting design science within the Information Systems (IS) discipline (Australian National University, Australia et al., 2020; Brocke et al., 2020; Venable et al., 2017). DSRM bridges the gap between IS, a multidisciplinary applied research discipline that creates solutions for organizations using IT, and engineering, for its strong focus on system design and analysis to create practical solutions.

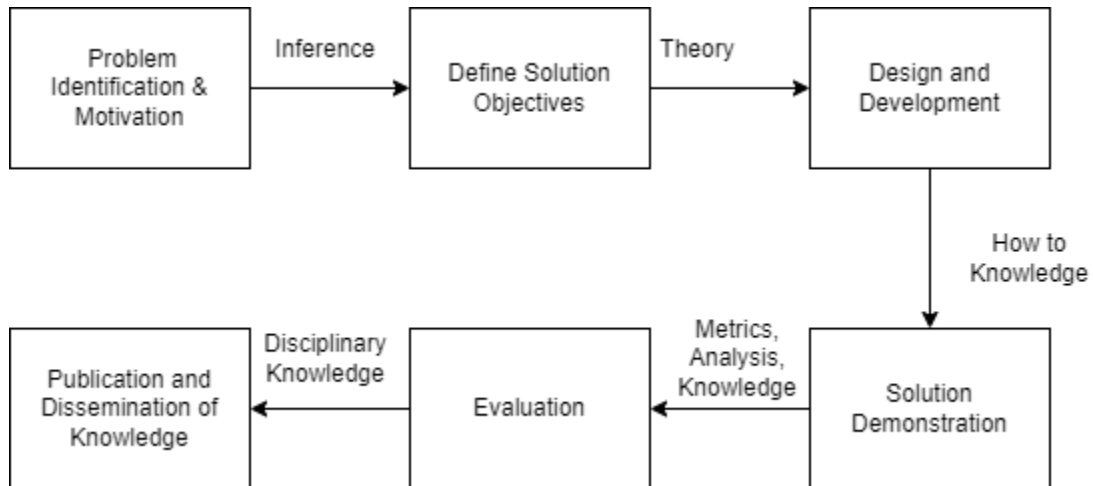


Figure 3 DSRM Process Model (Peffers et al. 2007)

DSRM's Process Model revolves around six main research activities, as shown in Figure 3. The first activity is identifying the problem and discussing the motivations for solving the problem. A well-defined problem creates a clear path to developing an effective solution. Moreover, justifying the value of the research solution motivates both the researcher and the target audience to continue moving forward toward the solution and accepting its results.

Once the problem has been identified clearly and motivations have been established, researchers can progress to the second activity, defining the solution objectives. Researchers with knowledge of the problem can infer and interpret the objectives of the solution and determine whether the solution is possible and feasible. The objectives can be quantitative or qualitative. Researchers can initiate the third activity, design and development, with a comprehensive background and knowledge of the solution. The design research artifact must have its research contribution embedded into its design. Researchers should clearly envision the design artifact's functionality and architecture which can be demonstrated and measured.

The fourth activity in the DSRM process is Demonstration, where the researchers provide the necessary evidence that their design artifact addresses and solves the

original research problem. Demonstration methods include setting up experiments, conducting simulations, providing case studies, or offering proof of solution.

The penultimate activity is Evaluation, where researchers observe and measure how successfully the design artifact solved the research problem. Evaluation can be done in an array of different approaches and techniques. For example, we can compare the artifact functionality with the solution objective to see how well-aligned they are to each other from a qualitative perspective. From the quantitative perspective, researchers can utilize performance measures such as survey results, clients' feedback, and tangible results such as items produced or increased or decreased expenditures on a project. Another example would be measuring the performance of a system, with metrics such as response latency, availability, and scalability, amongst other metrics. Overall, the evaluation should provide appropriate empirical evidence and logical proof to convince the academic and practitioner audience that are interested in the research problem. Depending on the nature of the research, the evaluation may lead to researchers go back to design and development to improve their design artifact, or to recommend future enhancement or research trends.

The final activity in the DSRM process is Communication through the publication and dissemination of knowledge. At this stage of the process, the researchers would have gained immense knowledge and insight into the research problem and its subsequent solutions based on their experience in designing and developing the design artifact. In a scholarly environment, researchers could use the structure of this process to structure their research publication (research problem, literature review, hypothesis development, data collection and analysis, results, discussions, and conclusions). As for practitioners, communication can be in the form of articles, blog posts, and keynote presentations to communicate the research problem's importance and the research solution's novelty and effectiveness.

This thesis will apply the DSRM research methodology to create and implement the Blockchain-based Intelligent Automation (BCIA) bot framework. Furthermore, to demonstrate the BCIA bot’s functionality and ability to create secured activity logs and payment receipt evidence on the BC, the researcher will run a simulation of a small business processing invoices through the BCIA bot. The evaluation phase will be discussed in detail in Chapter Four. Figure 4 illustrates the DSRM process model for this thesis.

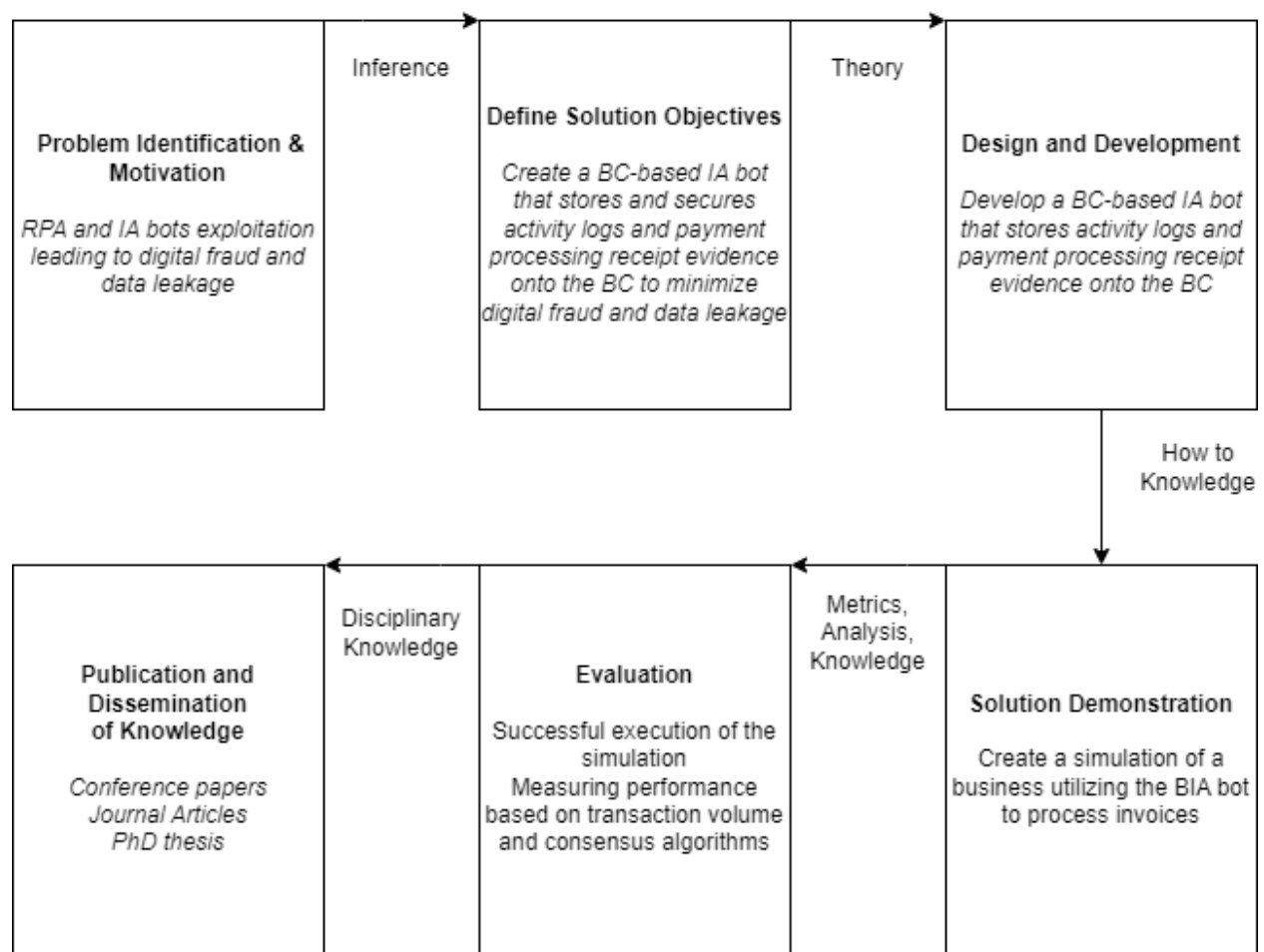


Figure 4 BCIA Bot Framework DSRM Process Model

3.3 Problem Identification and Motivation

This section presents two real-world scenarios highlighting the cybersecurity risks associated with RPA and IA bots to identify the research problem and its motivation:

Data Leakage: Data leakage can result from weak access controls, insider threats, or hacking. As shown in Figure 5, a malicious actor, possibly a disgruntled employee, a rogue bot developer, or an attacker, gains unauthorized or illegal access to the IA bot by obtaining its login credentials. Since the IA bot handles customers' private data, the malicious actor can hack, access illegally, or abuse the bot's credentials to either reprogram the bot to transmit specific data outside the organization to a third party or override the bot manually and leak the data. Most cybercriminals know to hide and cover their tracks by accessing the bot's activity logs, either by tampering with them and modifying their content or by deleting the logs altogether, with the prior less suspicious.

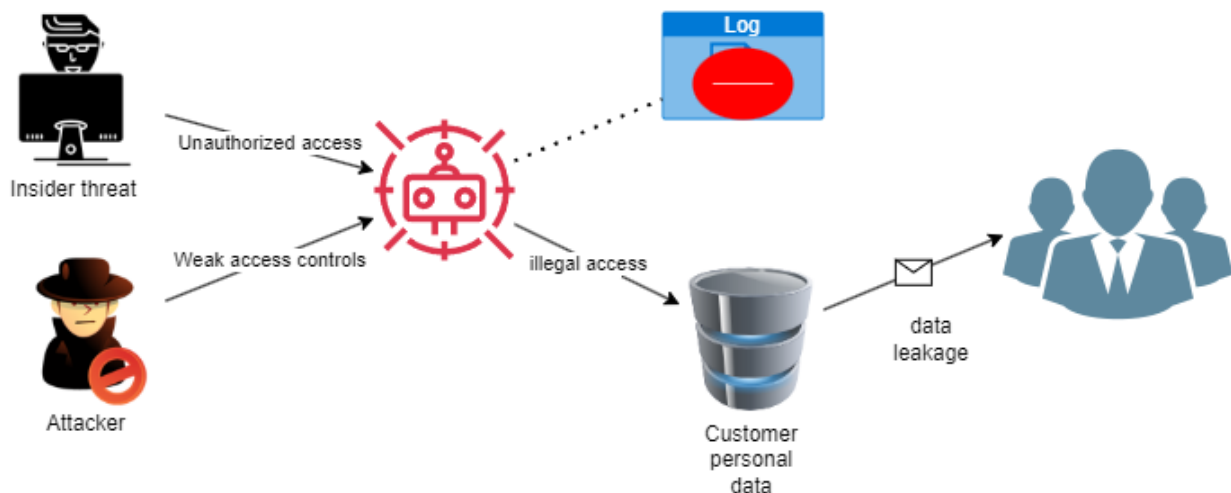


Figure 5 Data Leakage Scenario

Digital Fraud: In this scenario, the malicious actor, most likely an insider threat, gains unauthorized or illegal access to the bot's login credentials. As shown in Figure 6, the malicious actor begins to tamper with the IA bot's automated processes. He modifies the bot's instructions to divert funds from customer's transactions into his cryptocurrency wallet, exploiting vulnerabilities in the payment processing system. To evade being caught, the malicious actor manipulates or deletes the bot's activity log and all evidence of the payments forwarded to the criminal's cryptocurrency wallet.

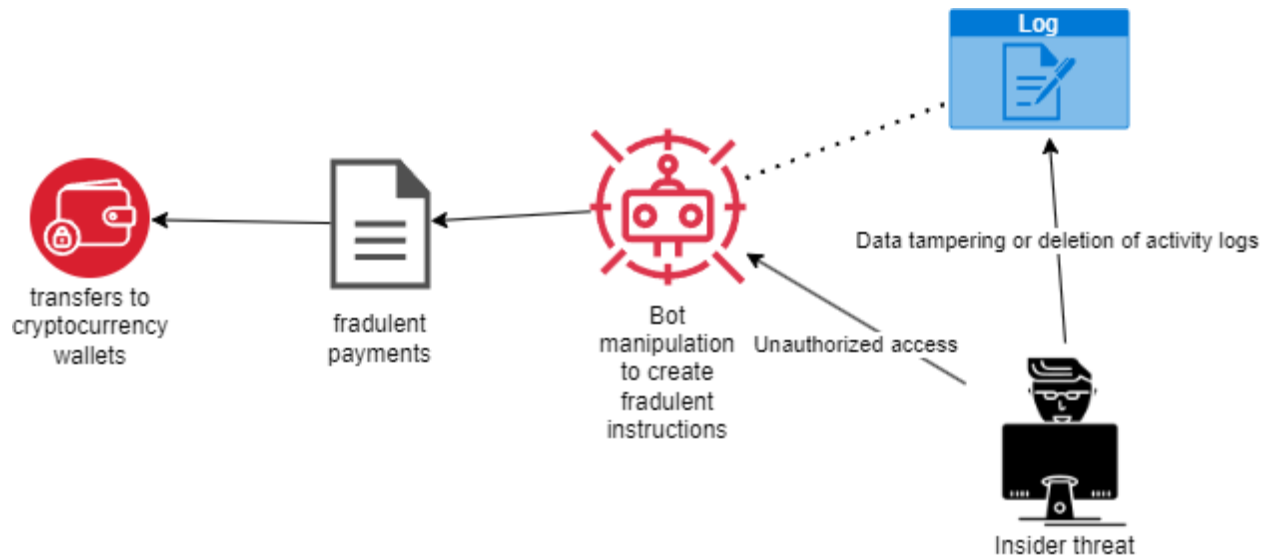


Figure 6 Digital Fraud Scenario

3.4 Design and Development: The BCIA Bot Framework

The BCIA Bot Framework has two primary purposes to overcome the security challenges of Data Leakage and Digital Fraud. The first is the IA invoice processing bot, which predominately follows the typical workflow for RPA invoice processing as described in Sahu et al. and Kerutis and Cakneryte's works (Kerutis & Calneryte, 2022b; Sahu et al., 2020). The second purpose is the utilization of BC technologies

to create and store copies of the bot's activity logs and payment receipts onto the BC ledger to maintain the logs' and payment receipt evidence integrity and functionality

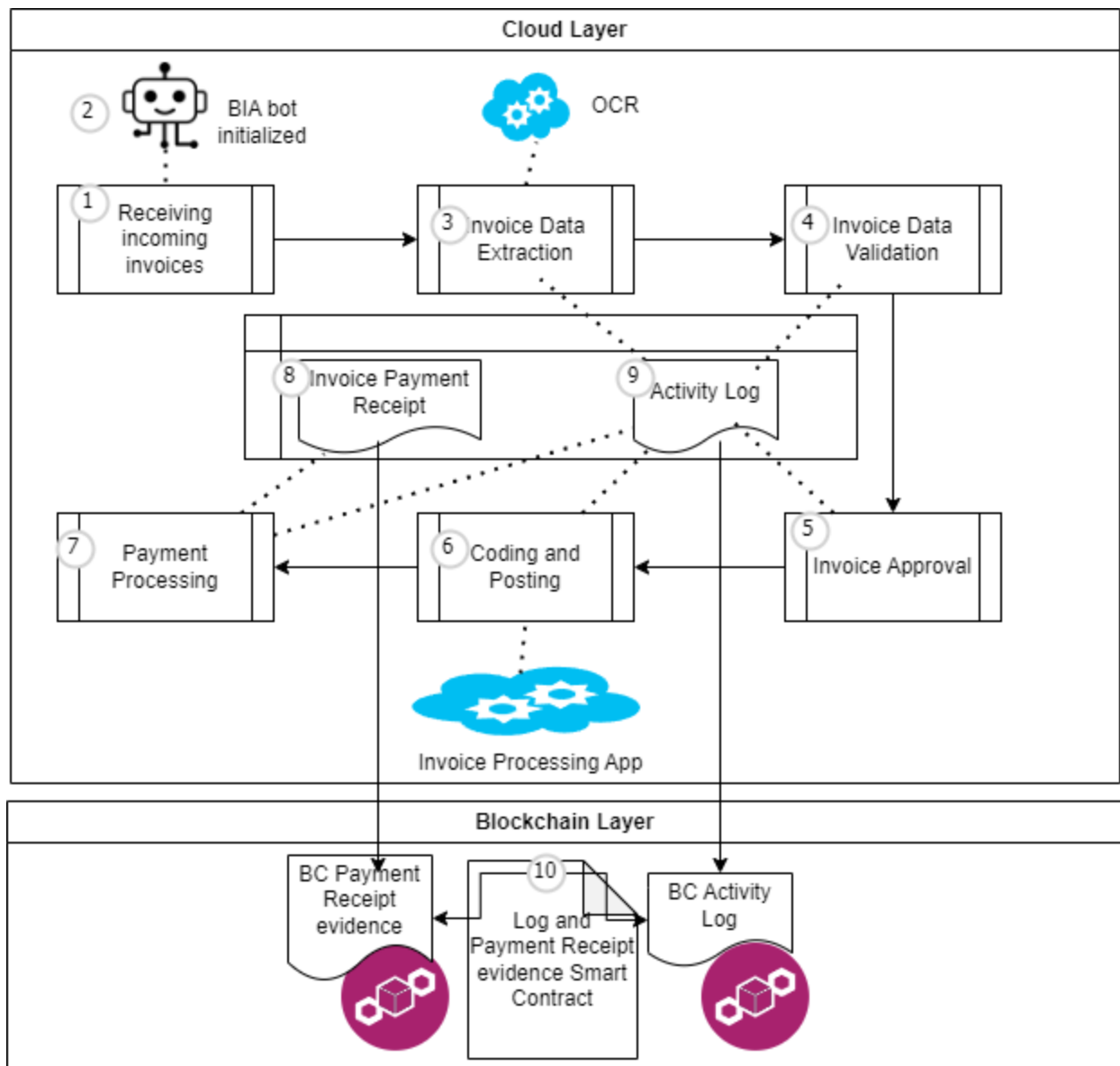


Figure 7 The BCIA Bot Framework

as digital evidence in the case of a digital forensics investigation. The workflow as shown in Figure 7 comprises ten steps: (1) Receiving incoming invoices, (2) BCIA bot initialization, (3) invoice data extraction, (4) invoice data validation, (5) invoice approval, (6) costing and posting, and (7) payment processing. Each step is logged into an activity log stored in the computer's internal storage as shown in step (9),

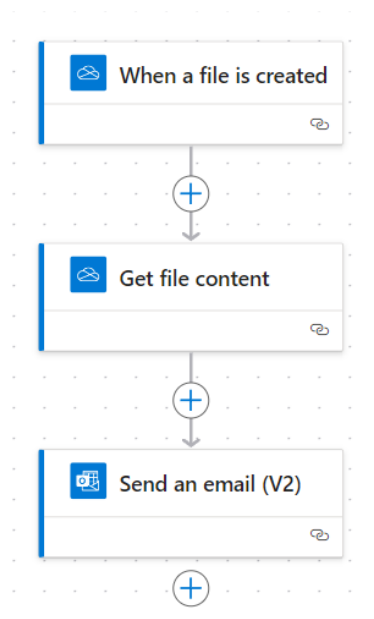
with a copy generated and stored on a BC ledger in step (10). Once the payments are processed and an invoice payment receipt has been created in step (8), another copy is generated and stored on to the BC ledger, similarly to the activity log's process in step (10). The BCIA bot leverages ML-based text extraction algorithms using Optical Character Recognition (OCR) through Textractor and seamless integration with an invoice processing application, Invoice Ninja, and its API on an AWS EC2 server. Textractor is a document intelligence service created on Python that works seamlessly with Amazon Textract and specializes in text recognition, table extraction, and form processing (*Textractor Documentation — Amazon-Textract-Textractor 1.0.0 Documentation*, 2022). Invoice Ninja is a Software-as-a-Service (SaaS) platform designed to handle SMEs businesses' invoice processing requirements. The following subsections will describe every step in the BCIA Bot Framework's workflow in further detail.

3.4.1 Receiving incoming invoices

Firstly, the small or medium-sized enterprise (SME) receives the vendor's invoice in PDF format through the BCIA bot's email inbox, as many SMEs generally have limited resources and cannot afford high-end ERP systems to manage their invoices (Haddara & Zach, 2011) and PDF is a widely accepted formal document format. The bot's automation process begins when a PDF attachment in the incoming email is detected. Since invoices store many personal data items such as names, addresses, and payment details, there are no readily available invoice datasets online. Therefore, the researcher generated 1000 dummy invoices utilizing six distinct templates denoting different companies and invoice structures to simulate a real-world situation. The templates were populated with random invoice data using macros and Mail Merge in Microsoft Office for the experiments.

Figure 8 Simple Power Automate workflow to send emails to the BCIA Bot

Over 1000 invoices were sent to the BCIA bot's email address by creating a simple automation process in Microsoft's Power Automate as shown in Figure . The automated cloud flow is triggered by adding the invoice PDF files in a designated OneDrive folder. Next , each invoice PDF file has a unique filename that will be used as an email attachment. The final step of the cloud flow is composing and automating the email messages to be sent to the BCIA bot's email. The email message comprises a Subject line with the filename of the invoice PDF, a pre-written message body, and the invoice PDF file as its email attachment. Below is the algorithm for how the BCIA bot handles invoice emails.



Algorithm 1: Email Processing Workflow

Input: Gmail API credentials, Scope for Gmail API, Query string for email search

Output: Processed email attachments saved locally, Logs detailing the processing steps

1: Initialize Logging

2: Initialize Google API client using OAuth 2.0 credentials

-
- 3: Create Gmail API service using the specified API name, version, and scopes
 - 2: Define Custom Exception Classes: GmailException, NoEmailFound
 - 3: Search emails matching the query string (emails with PDF attachments)
 - 4: Retrieve attachment data from email messages
 - 5: Retrieve detailed metadata about email messages
 - 6: Process email attachments, saving them locally and logging the process
 - 7: Execute the script , triggering the email process workflow
-

3.4.2 Invoice Data Extraction

Once the email message is received in the BCIA bot's inbox, the bot is initialized and checks whether a PDF file is attached. In order to automate the data entry from the PDF invoice into the invoice processing software, the bot uses Optical Character Recognition (OCR) technology through the Textractor¹ python package, which connects seamlessly with Amazon Web Services (AWS)'s Textract API to extract the necessary invoice data from the PDF file to post into the invoice processing system. Table 7 shows that the extracted invoice data include the vendor company name, vendor address, date, invoice number, purchase order number, items, and payment data. The algorithm of this step in the workflow can be found in Algorithm 2: Steps 4 and 5.

Table 7 Invoice data extraction fields

No.	Invoice Data Field	Data Type
1	Vendor Company Name	Text
2	Vendor Address	Text
3	Date	Date
4	Invoice Number	Data

¹ <https://aws-samples.github.io/amazon-textract-textractor/index.html>

5	Purchase order number	Data
6	Items	Text
7	Payment data	Float

3.4.3 Invoice data validation and approval

The following step in the workflow is validating the invoice data by comparing the extracted data with the Purchase Order (PO) numbers, where the process continues whenever the PO number and the invoice number match. In some cases, the OCR may have some errors; therefore, having pre-defined validation rules and formats is necessary to ensure data accuracy.

Once the invoice data is validated, attended RPA bots would need to wait for a human user to approve the invoice and give the green light to continue the transaction. As this IA bot is unattended by design, the approval step is automated directly after being validated in the previous step. The algorithm of invoice data validation in the BCIA Bot Framework can be found in Algorithm 2: Step 6.

3.4.4 Coding and posting

Invoice processing usually happens through an Enterprise Resource Planning (ERP) system, but due to the limitations of this research, an open-source invoice processing application handles the coding and posting. The approved invoice data is sent to the invoice processing software, Invoice Ninja², to be coded and posted. Coding involves assigning specific identifiers or codes to invoice elements, such as product items and prices, to categorize and track them. Once the data is coded, the next step is posting, where the BCIA bot enters the invoice data correctly into the ERP system

² <https://invoiceninja.com/>

and stores it based on the previous coding. Posting the invoice finalizes the transaction preparation for payment processing and makes it legally binding between buyer and seller. Posting the invoice prevents anyone from deleting the invoice or tampering with any data in the invoice. The algorithm of this step is found below.

Algorithm 2: Automated Invoice Processing with Textract and Invoice Ninja API

Input: PDF files directory, Invoice Ninja API URL, API Token, AWS

Credentials: Access Key ID, Secret access key, and region for AWS Textract service, Environment Variables .env file

Output: Created Invoices: Successful creation of invoices logged, Error messages

1: Load Environment Variables: .env file, Textract API credentials, file paths

2: Initialize Textractor and API configuration: Configure Textractor with AWS credentials, Set up Invoice Ninja API authentication with the provided token and URL.

3: Define Helper Functions: Functions for interacting with the API (creating invoices, checking clients, downloading PDFs)

4: Main Processing Loop: Iterate over each PDF file in the specified directory.

- For each PDF file:
 - Extract text and structured data using Textractor
 - Parse extracted data to retrieve invoice information.
 - Create a dictionary with extracted data and additional information
 - Use API functions to create an invoice with the extracted data.
 - Download the processed PDF for record-keeping
-

5: Extracting Information from PDFs: Use Textractor to extract structured data from PDF files, parse the extracted data to identify and extract relevant fields such as vendor information, invoice dates, line items, etc.

6: Creating Invoices:

- Check if the client/vendor exists in the system using the API,
- If the client exists retrieve their Client ID, if not, create a new client/vendor using the provided data.
- Use the extracted invoice information to create an invoice via the Invoice Ninja API, associating it with the correct client/vendor.
- Handle additional invoice details such as tax rates and due dates.

7: Error Handling and Logging: Implement error handling for API requests, file operations, and data extraction, Log important events, errors and debugging information for monitoring and troubleshooting.

3.4.5 Payment processing

Payment processing is the final step in the typical RPA invoice process, where payments can be scheduled or processed automatically through the invoice processing application. Once processed, the application generates payment confirmations and update the system's relevant records.

As shown in Figure 8, the BCIA bot's activity logs store the timestamps and actions of each of these six steps on the local storage. The data stored in the activity logs are the timestamp, activity, username, system, node, release, machine, and processor details. This stored information will clearly show any abnormalities in bot's behavior or remote access.

```

2024-02-22 17:46:50,787 - file_cache is only supported with oauth2client<4.0.0
2024-02-22 17:46:57,001 - File invoice4.pdf is saved at C:\Users\catch\Desktop\BCTEST/files/; System Info: uname_result(system='Windows',
2024-02-22 17:47:08,167 - Data Extracted; System Info: uname_result(system='Windows', node='NEBULA', release='10', version='10.0.19041',
2024-02-22 17:47:08,170 - Excel File Created; System Info: uname_result(system='Windows', node='NEBULA', release='10', version='10.0.1904
2024-02-22 17:47:09,418 - File invoice3.pdf is saved at C:\Users\catch\Desktop\BCTEST/files/; System Info: uname_result(system='Windows',
2024-02-22 17:47:20,356 - Data Extracted; System Info: uname_result(system='Windows', node='NEBULA', release='10', version='10.0.19041',
2024-02-22 17:47:20,357 - Excel File Created; System Info: uname_result(system='Windows', node='NEBULA', release='10', version='10.0.1904
2024-02-22 17:47:21,222 - File invoice2.pdf is saved at C:\Users\catch\Desktop\BCTEST/files/; System Info: uname_result(system='Windows',
2024-02-22 17:47:32,079 - Data Extracted; System Info: uname_result(system='Windows', node='NEBULA', release='10', version='10.0.19041',
2024-02-22 17:47:32,080 - Excel File Created; System Info: uname_result(system='Windows', node='NEBULA', release='10', version='10.0.1904
2024-02-22 17:47:33,081 - File invoice1.pdf is saved at C:\Users\catch\Desktop\BCTEST/files/; System Info: uname_result(system='Windows',
2024-02-22 17:47:44,000 - Data Extracted; System Info: uname_result(system='Windows', node='NEBULA', release='10', version='10.0.19041',
2024-02-22 17:47:44,001 - Excel File Created; System Info: uname_result(system='Windows', node='NEBULA', release='10', version='10.0.1904
2024-02-22 17:47:44,932 - File invoice (1230).pdf is saved at C:\Users\catch\Desktop\BCTEST/files/; System Info: uname_result(system='Win

```

Figure 8 Example of BCIA bot activity log

Once an activity log is created, it will simultaneously store a copy on the BC logging system, thus enabling the system to have a tamper-free audit trail to detect any malicious actions conducted through unauthorized access and use of the bot as shown in **Error! Reference source not found.**

3.4.6 Adding Log entries from File to BC

As previously shown in **Error! Reference source not found.**, a copy of the BCIA bot's activity logs are securely stored on the BC. Whenever an investigation requires access to a particular log and timestamp, a search query can be made through a simple search engine, and because the logs are on the BC, the files are immutable and read-only. Therefore, the BC log ledger becomes an extremely valuable evidence depository whenever suspicious activity occurs in the BCIA bot or within the invoice processing application. Below is the algorithm explaining the steps to adding the long entries for the activity log on file to the BC.

Algorithm 3: Add Log Entries from File to Ethereum BC

Input: Ethereum address of the smart contract, private key associated with the Ethereum account used to send transactions, Path to the log file containing entries to be added, gas price in Gwei for transactions, Timestamp for retrieving a specific log entry

Output: transaction hash of the added log entry (if successful), Retrieved log entry from the BC

1: Connect to Ethereum Network: Connect to the Ethereum network via a testnet using a provider like Infura.

2: Set up Ethereum account: Load the private key associated with the Ethereum account, create an account object from the private key

3: Load Contract Address and ABI: Define the contract address and ABI necessary for interacting with the SC

4: Define Helper Functions: Check if log entry exists, adding a new log entry to the BC, retrieve a log entry from the BC.

5: Create Contract Instance: Create an instance of the smart contract using the contract address and ABI

6: Read Log file: Open the specified log file path for reading, Iterate through each line in the log file

7: Process Log Entries: For each line in the log file:

- Extract the timestamp and message
- Check if a log entry already exists for the timestamp
- If the log entry does not exist, add it to the BC
- Wait for the transaction to be mined and retrieve the transaction hash

8: Retrieve Log Entry (Optional): If timestamp to retrieve is provided, retrieve the log entry from the BC

9: Output results: Print the transaction hash of added log entries, Print the retrieved log entry if timestamp to retrieve is provided.

3.4.7 Logging payment receipts onto the BC

Algorithm 4 describes the process of creating and storing the invoice payment evidence on the BC.

Algorithm 4: Payment Log Smart Contract

Data Structures

- PaymentInfo: a structure that holds:
 - ‘data’: A string containing payment information in JSON format
 - paymentLogs: A mapping that associates a string ‘paymentId’ with ‘PaymentInfo’

Functions:

1: addOrUpdatePaymentLog (paymentId, paymentInfo)

- Inputs:
 - ‘paymentId’ (string): The unique identifier for the payment entry.
 - paymentInfo’(string): The payment details in JSON format.
 - Process:
 - Check if ‘paymentId’ exists in ‘paymentLogs’ by evaluating the length of ‘data’ for the ‘paymentId’.
 - If ‘paymentId’ does not exist (i.e., a new entry):
 - Store ‘paymentInfo’ in ‘paymentLogs’ under the key ‘paymentId’.
 - Emit an event ‘PaymentLogAdded’ with ‘paymentId’ and ‘paymentInfo’
 - If ‘paymentId’ already exists:
-

-
- Update 'paymentInfo' in 'paymentLogs' under the key 'paymentId'
 - Emit an event 'PaymentLogUpdated' with 'paymentId' and 'paymentInfo'
- Outputs: None, the function modifies the blockchain state and potentially emits events but does not return any value.

2: getPaymentLog(paymentId)

- Inputs: 'paymentId' (string): The unique identifier for the payment entry to be retrieved
 - Process:
 - Check if 'paymentId' exists in 'paymentLogs' by evaluating the length of 'data' for the 'paymentId'.
 - If not data exists for the 'paymentId', revert the transaction with an error message "Payment ID not found".
 - If data exists, retrieve 'paymentInfo' from 'paymentLogs' using 'paymentId'.
 - Outputs: Returns the 'paymentInfo' associated with 'paymentId' as a string, if the 'paymentId' does not exist, the function execution fails with an error.
-
-

Next, the primary objective of the script is to facilitate the integration between the BC network, and Invoice Ninja invoice processing system to complete the payment processes. This integration allows for the seamless transfer of payment information between the two systems while maintaining a record of transactions on the BC. The algorithm is described below.

Algorithm 5: Blockchain Payment Integration

Inputs:

- Environment variables from .env file:
 - InfluxDB host, port, user, password, database
 - Invoice Ninja API URL, API Token, API secret
 - Private key
- Ethereum blockchain contract details:
 - Contract address
 - Contract Application Binary Interface (ABI)
- External API endpoint for fetching payments

Output: Addition of payments to the BC , recording of BC metrics in InfluxDB

1: Import necessary libraries and modules: dotenv, requests, os, json, datetime, web3, eth_account, InfluxDBClient.

2: Load environment variable from the .env file.

3: Set up connections: Connect to InfluxDB using credentials (host, port, username, password, database, and connect to Ethereum network using Infura (or another provider).

4: Define the Ethereum BC smart contract: specify the contract address and ABI.

5: Define functions for interacting with the BC smart contract:

- Fetch payments from Invoice Ninja API: Input: API token, API URL, Output: List of payment data
 - Check if a payment exists on the BC: Input: Payment ID, Output: Boolean to indicate existence of payment.
 - Add a payment to the BC: Input: Payment ID, Payment Information, Output: Transaction hash, Confirmation time
-

-
- Retrieve payment information from the BC: Input: Payment ID, Output: Payment information or NONE
 - Push BC metrics to InfluxDB: Input: Metrics, latest block number Output: None.

6: Fetch payments from the Invoice Ninja API: call the fetch payments function to retrieve payment date.

7: Process payments: For each payment retrieved: Check if the payment exists on the BC, if not, add the payment to the BC and update metrics in InfluxDB.

8: Calculate blockchain metrics periodically:

- Calculate transaction throughput, network hashrate, latest block difficulty, transaction count, and block size.
- Push the metrics to InfluxDB

9: Handle errors and exceptions: Implement error handling mechanisms for API requests and blockchain transactions.

10: Main execution: Execute fetch payments, process them, and update metrics.

The implementation of the BCIA Bot Framework successfully integrated BC technology with the invoice processing application, enabling the secure recording and verification of payments on the Ethereum blockchain. Through smart contracts, payments are added to the blockchain in a transparent and immutable manner, enhancing trust and reliability. Additionally, the implementation collects and analyzes blockchain metrics to monitor network performance and transaction throughput, providing valuable insights for system optimization and scalability. Overall, the script bridges an external payment system and the Ethereum blockchain,

facilitating the seamless transfer of payment information and recording blockchain metrics for monitoring and analysis.

3.5 BCIA Security Architecture

The security architecture of the BCIA Bot Framework follows the architecture of Block-DEF (Tian et al., 2019) and BCFL Logging (Awuson-David et al., 2021), which are both BC-based secure digital evidence frameworks. The shared similarities are that all three frameworks aim to create tamper-proof evidence by utilizing BC technologies, which can be retrieved for digital forensic investigations. In contrast, as shown in Table 8 **Error! Reference source not found.**, the BCIA Bot Framework relies on a public BC network to create the necessary evidence information. It also utilizes AWS cloud technology as a service layer to create the evidence and as the network layer to interact with the public BC. The experimentation in Chapter Four will determine if utilizing public BC networks can achieve the same results as the Block-DEF and BCFL logging frameworks.

Table 8 Comparison between Block-DEF, BCFL logging and BCIA Frameworks

Security Framework	Architecture	Blockchain type	Consensus Algorithm
Block-DEF	Three-Layer: • Service • Network • Blockchain	• Private (permissioned) • Consortium (hybrid)	• Name-based PBFT • Raft
BCIA Framework	Two-Layer: • Cloud service • Blockchain	Public (permissionless)	• Sepolia ETH (PoS)
BCFL Logging Framework	Three-Layer: • Supply chain layer • IaaS Cloud layer • Blockchain layer	• Hyperledger Fabric (permissioned)	• BFT

3.6 Summary

This chapter introduced the methodology used in the thesis, DSRM, which was chosen for its utility in creating and evaluating IT artifacts. It explained the adoption of DSRM to address the thesis’ needs and outlines two key RPA/IA-related cybersecurity scenarios—data leakage and digital fraud—that provide context for the proposed solutions. The BCIA Bot Framework is introduced as a novel approach to combating these challenges. Detailed steps of the BCIA Bot Framework include an invoice processing workflow involving data extraction, validation, coding, and payment processing. The integration of blockchain technology is explained, highlighting its role in logging and validating transaction data to ensure security and

integrity. Additionally, the chapter outlines the BCIA Security Architecture, including the Block-DEF and BCFL Logging Framework, designed to secure transaction data against unauthorized alterations. The practical application of the BCIA Bot Framework to real-world use cases demonstrates its effectiveness in a controlled environment, thereby providing a comprehensive overview of its potential benefits in enhancing security measures within digital transactions.

Chapter 4 Experimental Results and Discussion

4.1 Overview

This chapter will describe the experiments implemented on the BCIA Bot Framework to answer RQ2 and RQ3. The experiment created is a case study scenario where a small business decided to use the BCIA Bot Framework to process its incoming invoices. Firstly, to determine its success in creating the activity logs and payment process receipt evidence on the BC, and then conduct several experiments to measure the performance of the BCIA bot in processing the invoices and evidence generated utilizing three BC testnets: Sepolia ETH, Solana Devnet, and BNB Chain. The following sections in this chapter will discuss the testnets used in the experiments, the invoice dataset used, and how the experiment iterations are set up. The performance metrics: Network Throughput, Block size, Transaction fees, Transaction Latency, and Finality Time are also discussed in detail. Next, the results of each experiment's iteration are presented, and the chapter closes with a discussion of the results.

4.2 Testnets And Performance Metrics Used In The Experiment

1. **Sepolia ETH:** This is a testnet for the Ethereum network, which has transitioned to a Proof of Stake (PoS) mechanism following the Ethereum 2.0 upgrade. Therefore, Sepolia operates under a PoS consensus model.
2. **Solana Devnet:** Solana created a unique consensus mechanism called Proof of History (PoH), combined with a Proof of Stake (PoS) variation. Proof of History helps to create a historical record proving that an event has occurred

at a specific time. This is combined with PoS to select the network's security and validator.

3. **BNB Chain:** Previously known as Binance Chain and Binance Smart Chain, which have now merged into what is called BNB Chain, has recently shifted to PoSA from DPoS (*Introduction | BNB Chain Documentation*, 2024). The primary motivation for this shift is to combine the characteristics of DPoS and PoA so that blocks are produced by a small set of validators, who take turns in block production similar to PoA.

This section will present and define the five performance metrics utilized in this thesis' experiments. Based on the works of (Bamakan et al., 2020; Hattab & Taha Alyaseen, 2019; Huang et al., 2020; Mingxiao et al., 2017), the shared and agreed upon metrics were as followed: Network Throughput, Block size, Transaction fees, Transaction Latency, and Finality Time.

Network throughput refers to the number of transactions a BC network can process in a given timeframe. It is considered a critical performance metric that affects how quickly and efficiently a BC network can perform and operate without failures. Block Size describes the maximum amount of data that each block in the BC can hold and is seen as a key factor in the design of a BC network as it affects the scalability and speed of the network.

Transaction fees, measured in cryptocurrency or tokens, are the costs of processing transactions in a BC network. Transaction fees are small amounts of cryptocurrency users pay to compensate for the computational energy needed to process and validate transactions on the network. Transaction latency is the time between initiating a transaction or payment and receiving confirmation that it is valid. In classic BFT systems, a transaction is finalized once it gets confirmed, whereas, in longest chain

consensus like Solana and Ethereum PoS, a transaction may get included in a block and the re-orged. As a result, we need to wait until a transaction is “k-block deep,” resulting in a significantly greater latency than a single confirmation. (*Understanding Blockchain Latency and Throughput*, 2022). Transaction Latency can be calculated as the time that a transaction was added into a block minus the time that the transaction was submitted to the network (Klaytn, 2022)

Table 9 Average block generation time and Finality time for Solana, BNB Chain, and ETH

Blockchain	Block generation time (secs)	Finality time
Solana	0.4	12.8 second
BNB Chain	3	7.5 seconds
ETH	12	~16 minutes

Finality Time is the duration of time required for a transaction to achieve finality within a blockchain network. Finality refers to the point at which a transaction becomes immutable in the BC’s ledger (Chainspect, 2024). This metric is essential for networks where quick finality is critical for application performance. Table 9 displays the average block generation time and finality time for Solana, BNB Chain, and ETH. There are three primary types of Finality time. Probabilistic Finality which the Bitcoin BC is a prime example of, the blocks don’t get completely finalized. However, they become more likely to be considered final over time. As time passes, the chance of a previous block being re-orged from the chain approaches zero, thus making the block effectively final. Provable Finality is when systems take steps to make finality happen faster and with stronger economic assurances. Provable Finality can be either instant or delayed provable finality. Delayed Finality chains

have better performance regarding responsiveness but are more complex and vulnerable to re-orgs. Examples are Ethereum and Solana (both delayed).

4.3 Dataset and Experiment Setup:

The dataset used for the experiment was created from six typical invoice templates found online via [invoicer.ai](#) (*Word Invoice Templates*, 2024) and populated with random data. Invoices by design and purpose contain sensitive personal data such as addresses, names, contact details, and bank account information; therefore, there is no publicly available online invoice dataset fit for this experiment. The total number of invoices generated was 1000 unique invoices.

The experiment will be conducted on three versions of the BCIA bot, and each bot will undergo four iterations to represent the invoice processing workloads of a small business, with 50 invoices and 100 invoices, and a medium-sized enterprise with 500 invoices, and 1000 invoices to simulate the workload of a large enterprise. These numbers are based on the work of (Touzon & Garuz, 2020).

The experiment aims to answer *RQ3: Which BC consensus algorithm provides optimal performance for BCIA bots for invoice processing?* Therefore, three versions of the BCIA bot were developed as each consensus algorithm and testnets interacts with different programming languages, virtual machines, and compilers, as shown in Table 10. For example, Ethereum’s SC programming language is Solidity, while Solana allows legacy programming such as C, C++, and Rust languages to write on-chain programs. More recently, Solana has added a Solidity compiler, Solang³, that allows developers to write Solana programs. Binance’s BNB Chain,

³ <https://github.com/hyperledger/solang>

on the other hand, is EVM-compatible and supports Solidity, GO, Java, and Swift programming languages.

Table 10 Testnet programming languages and execution environments

Blockchain Testnet	Consensus Algorithm	Programming Language(s)	Execution Environment	Purpose
Sepolia ETH	PoS	Solidity	Ethereum Virtual Machine (EVM)	Executing smart contracts
Solana Devnet	PoS PoH	C, C++ Rust Solang (Solidity Compiler)	Solana Virtual Machine (SVM)	Executing Solana programs
BNB Chain Testnet	PoSA	Solidity, GO, Java, Swift	BNB Beacon Chain	Governance (staking and voting)
			BNB Smart Chain	EVM compatible to execute smart contracts and protocols

4.4 Experiment Metrics Equations:

For the experiments, five primary BC network metrics are observed on each version of the BCIA bot, which will be labeled as ETH-BCIA for the Sepolia testnet-powered BCIA bot, SOL-BCIA for the Solana BCIA bot, and BNB-BCIA for the BNB Chain

powered bot. The metrics include Network Throughput, Block size, Transaction fees, Transaction Latency, and Finality Time, as previously discussed in this chapter. Network Throughput R , as shown in Equation (2), is measured in transactions per second (TPS) where B represents block size in bytes, A is the transaction size in bytes, and S represents the time intervals between blocks, measured in seconds (Hill et al., 2018).

$$R = \frac{B}{AS} \quad (2)$$

As for the block size B , it is the summation of all the individual transactions sizes included in the block where t_i is the size of the i -th transaction in the block, and n is the total number of transactions in the block. The block size cannot exceed the maximum block size limit M set by the BC's protocol $B \leq M$, as shown in Equation (3) below (Singh & Vardhan, 2020).

$$B = \sum_{i=1}^n t_i \quad (3)$$

Transaction fees are calculated differently depending on the BC, but generally follow the formula *Transaction Fee = Fee Rate $\times$ Transaction size* where the fee rate is the cost per byte or per unit of data and can fluctuate based on the network condition and congestion (*Comparing Blockchain Fee Structures*, 2023) . The Transaction size is measured in bytes as well. Ethereum, however, calculates the transaction fee based on gas, the unit of computational effort, where *Transaction Fee = Gas Used $\times$ Gas Price*.

Transaction Latency is measured in seconds and represents the time a transaction takes to be confirmed and recorded on the BC. Latency can be calculated using the following formula, as shown in Equation (4) (Catt, 2019):

$$Latency = T_{submit} + T_{mempool} + T_{block} + T_{propagation} + (N - 1) \times T_{block} \quad (4)$$

T_{submit} is the time to submit the transaction to the BC network, and $T_{mempool}$ is the time the transaction waits in the mempool. T_{block} denotes the average time to mine one block, and $T_{propagation}$ is the time taken for a block to be propagated to and accepted by the network. Finally, N is the number of confirmations required for the transaction to be become final.

Finality time is an important metric for DApps developers who require high transaction certainty to know how quickly and securely a transaction can be finalized and immutable on the BC ledger. Finality time is affected by several factors, including the CA and how blocks are validated, added to the BC, and finalized. Also, it is affected by network congestion and the BC networks' security requirements, which require multiple confirmations for security-sensitive applications.

Grafana Labs⁴ is the BC benchmarking tool used, and InfluxDB was utilized to store all the input data needed for the following experiments to measure the five metrics discussed in this section: Network throughput, Block size, Transaction fees, Transaction Latency, and Finality Time.

⁴ <https://grafana.com/>

4.5 Experiment 1: ETH-BCIA bot

The following experiments will simulate the BCIA bot's extraction of invoices, coding and posting, and processing the invoice payment, then creating the activity log and payment receipt, followed by the time it takes to create the invoice payment receipt evidence on the BC. The experiment is completed in increments, starting from 50 invoices to represent a small business, then 100 invoices, 500 invoices to simulate a medium-sized business, and finally, 1000 invoices to represent a large enterprise.

Small business simulation:

It took roughly 16 minutes to process a batch of 50 invoices using the Sepolia testnet, and Table 11 highlighted three significant increase/decrease points at the 13th, 15th, and 16th minutes, respectively. In total, there were 679 completed transactions.

Table 11 ETH-BCIA bot performance at 50 invoices

Minute After Start	0	13	15	16
Throughput Tx/Min	3.3	2.9	2.53	2.5
Transaction Counts	236	153	168	122
Block Size (bytes)	104073	193926	92041	217846

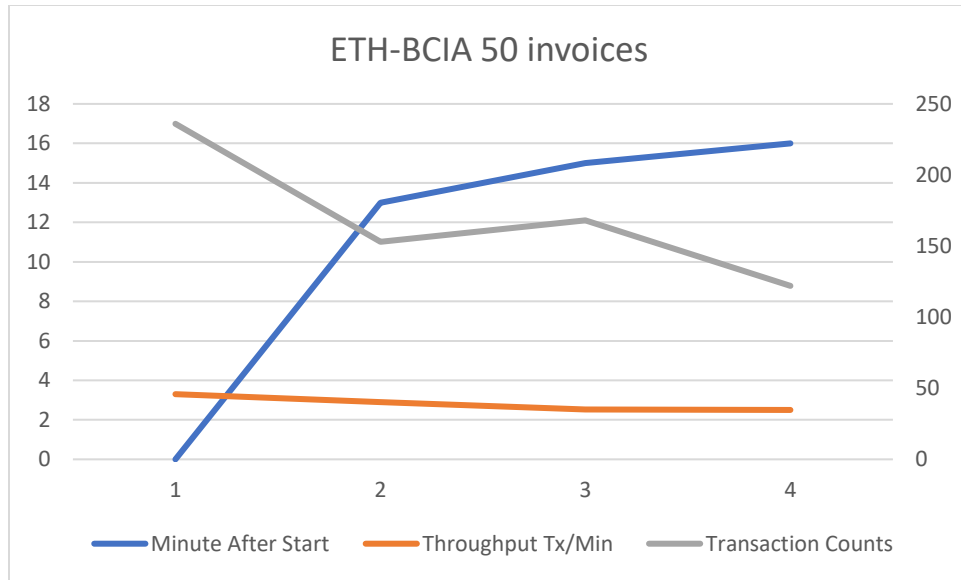


Figure 9 ETH-BCIA bot at 50 invoices

As shown in Figure 9, the number of transactions decreased steadily over the total time, with the throughput remaining reasonably steady.

At 100 invoices, there was a steady decrease in network throughput and the bot completed all the transactions within 20 minutes. Total completed transactions were 1354 transactions.

Table 12 ETH-BCIA bot performance at 100 invoices

Minutes after Start	Throughput tx/min	Transaction Counts	Block Size (bytes)
0	3.6	265	115632
6	3.2	173	205698
8	2.8	187	97542
11	2.7	132	236541
14	2.6	144	115569
17	2.4	156	2414558
19	2.2	149	1256983
20	2.1	136	2451447

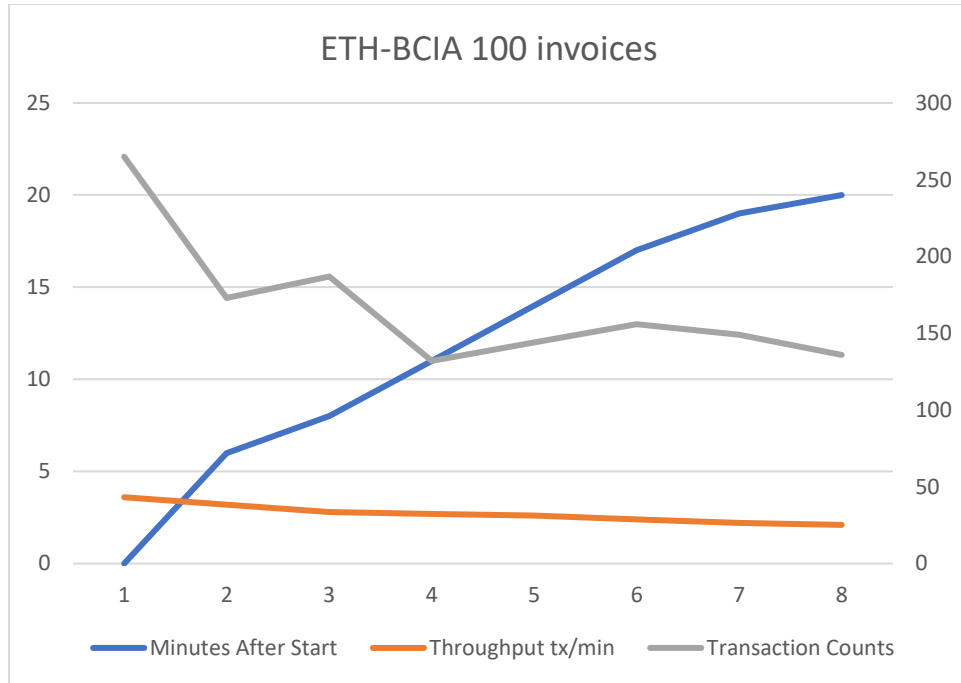


Figure 10 ETH-BCIA at 100 invoices

In **Error! Reference source not found.**, the initial throughput started higher than the previous iteration at 3.6 transactions per seconds and experienced a slight dip by the end of the process at 2.1 transactions per second.

Medium-sized business simulation: Although achieving a higher network throughput than the 50 invoice iteration at the start of the process, over time the throughput got lower and lost over 60% of its throughput by the end of the process, as shown in Table 13. In Figure 11, the network throughput decreases over 44% over time and the trend of decreasing number of transactions over time continues similarly to previous iterations (50 and 100 invoices).

Table 13 ETH-BCIA bot performance at 500 invoices

Minutes After Start	Throughput tx/min	Transaction Counts	Block Size (bytes)
0	3.8	1187	1040730
13	3.7	765	967315
15	2.6	842	460207
16	2.2	611	1089321
18	2	654	545263
21	1.9	789	1090587
24	1.8	707	600131
27	1.7	536	1112799

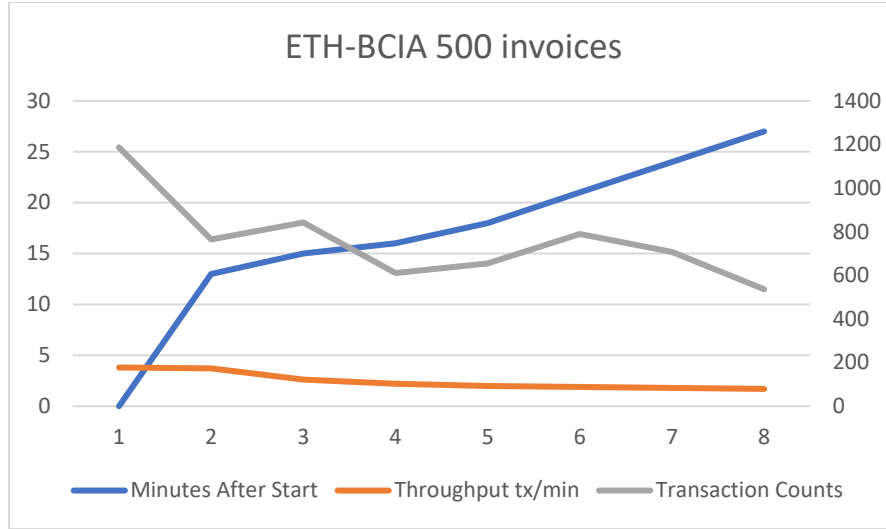


Figure 11 ETH-BCIA bot performance at 500 invoices

Large enterprise business simulation: As expected, the largest iteration of this experiment took the most prolonged period to complete all 1000 invoices, as shown in Table 14. In Figure 12, the performance trends are consistent with all three previous iterations. .

Table 14 ETH-BCIA bot performance at 1000 invoices

Minute After Start	Throughput tx/min	Transaction Counts	Block Size (bytes)
0	3	2394	1199823
7	2.6	1530	1599871
14	2.5	1684	1299963
16	2.3	1222	1499989
20	2.1	1308	899983
24	2	1578	1399937
26	1.9	1414	1199977
35	1.8	1072	999869
40	1.6	1160	899941
42	1.5	990	799981
48	1.4	830	699991
75	1.3	789	599963

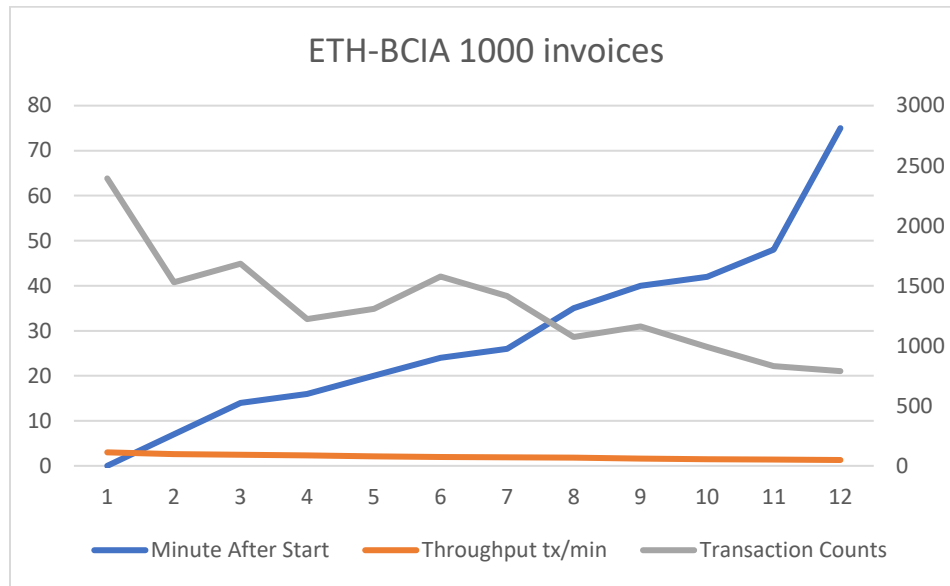


Figure 12 ETH-BCIA bot performance at 1000 invoices

Total transaction fees and finality times: as shown in Table 15, despite the fluctuations in block sizes, transactions, and network throughput throughout this experiment, the finality time remained consistent in all four iterations of invoice payment evidence creation. The transaction fees were also relatively proportional throughout this experiment, except for the 500 invoice iteration, where there was a significant dip in gas price despite all iterations being completed on the same day. Therefore, the gas prices did not fluctuate dramatically during that period. When converting the gas fees into \$USD, processing 50 invoices cost \$10.12, and 1000 invoices cost \$84.31.

Table 15 ETH-BCIA bots transaction fees and Finality Times

Number of processed invoices	Total transactions	Total Transaction fees	Transaction fees in \$USD	Finality Time (average)
50	679	0.0033993 ETH	10.12	75 secs
100	1354	0.0067268 ETH	20.03	79 secs
500	6091	0.013807 ETH	41.11	78 secs
1000	15971	0.02831665 ETH	84.31	75 secs

4.6 Experiment 2: SOL-BCIA Bot

Small business simulation: The first noticeable difference was that the throughput was exponentially higher than the ETH-BCIA Bot, and it took only three minutes to complete creating invoice payment evidence on the BC in a total of 1577 transactions as shown in Table 16, and 2750 transactions for 100 invoices as shown in Table 17. As shown in Figures 13 and 14, there is a noticeable increase in network throughput in comparison with the ETH-BCIA bot where the average throughput was 45 transactions per minute for the SOL-BCIA bot, whereas the ETH-BCIA had an average of 2.6 transactions per minute. This increase in throughput enabled the SOL-BCIA bot to process 50 invoices in under four minutes, and under six minutes to process 100 invoices.

Table 16 SOL-BCIA Bot performance at 50 invoices

Minutes after Start	Throughput tx/min	Transaction Counts	Block Size (bytes)
0	44	540	135677
1	46	633	174598
3	45	404	96322

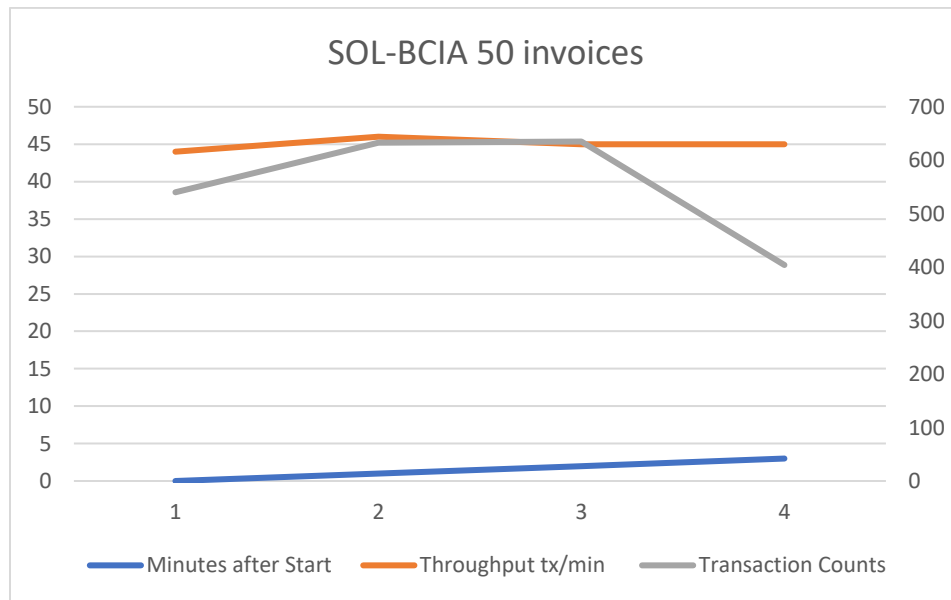


Figure 13 SOL-BCIA bot performance at 50 invoices

Table 17 SOL-BCIA Bot performance for 100 invoices

Minutes after Start	Throughput tx/min	Transaction Counts	Block Size
0	48	785	198542
1	46	723	192269
3	45	698	154786
5	47	544	151712

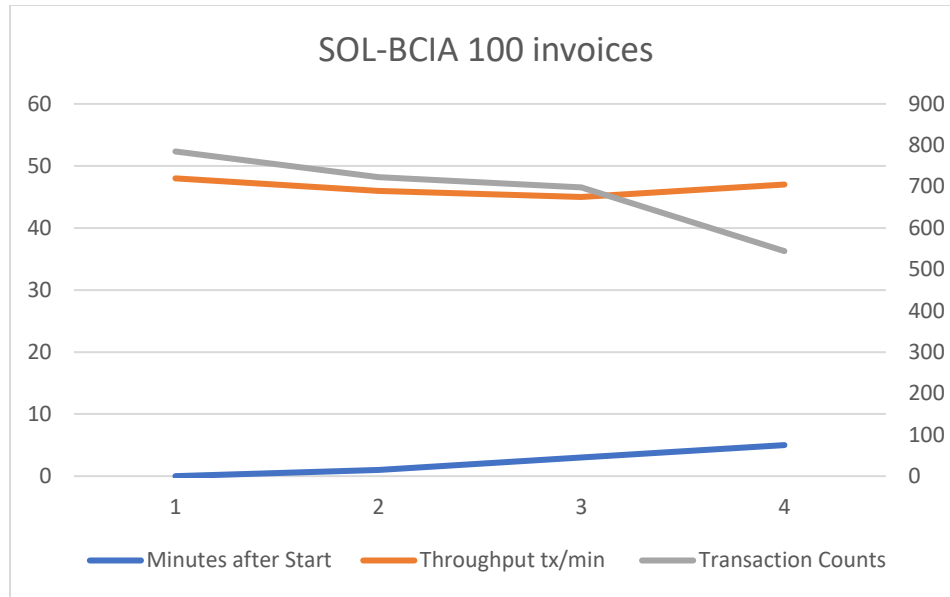


Figure 14 SOL-BCIA bot performance at 100 invoices

Medium Business simulation: At 500 invoices, the SOL-BCIA bot completed 6663 transactions within 13 minutes from the process start, halving ETH-BCIA bot's processing time. The throughput remained steady throughout the process, and the transaction counts lowered gradually until completion as shown in Table 18. Unlike the previous iterations, the transaction counts dipped dramatically over time starting from 2564 transactions at the beginning of the iteration and ending at 569, hence experiencing an 80% dip over time as shown in Figure 15.

Table 18 SOL-BCIA bot's performance at 500 invoices

Minutes After Start	Throughput tx/min	Transaction Counts	Block Size
0	45	2564	186522
2	44	1532	156698
5	43	1233	145839
7	45	765	168752
13	43	569	158742

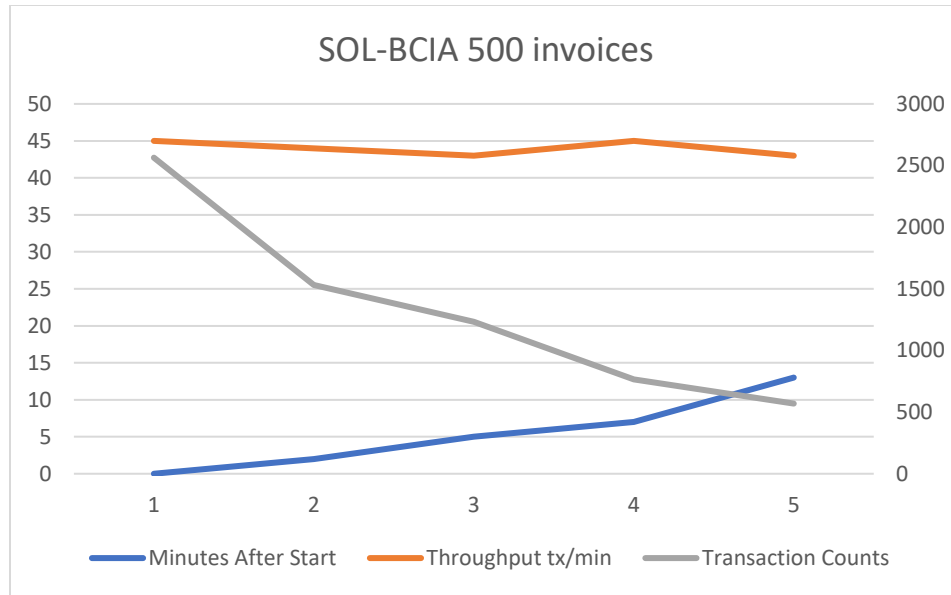


Figure 15 SOL-BCIA bot performance at 500 invoices

Large business simulation: Throughout the four iterations, the SOL-BCIA bot performed consistently with stable throughput and quick processing times compared to the ETH-BCIA bot's performance which was characterized by many fluctuations in transaction throughput, transaction counts, and block sizes. The process completed 57587 transactions on the BC network within 37 minutes, once again providing a faster speed than ETH-BCIA Bot as shown in Table 19. Also, as illustrated in Figure 16, the transaction counts fluctuated through the iterations experiencing two peaks and dips over 37 minutes.

Table 19 SOL-BCIA bot performance at 1000 invoices

Minute After Start	Throughput tx/min	Transaction Counts	Block Size
0	47	3354	157854
2	46	2547	198514
5	52	2254	162665
6	48	3221	169821
7	46	4899	163218
8	45	5448	125699
12	47	3877	114788
19	54	5849	358547
22	50	6954	289963
27	48	3669	157478
32	45	3548	185993
37	44	2241	187423

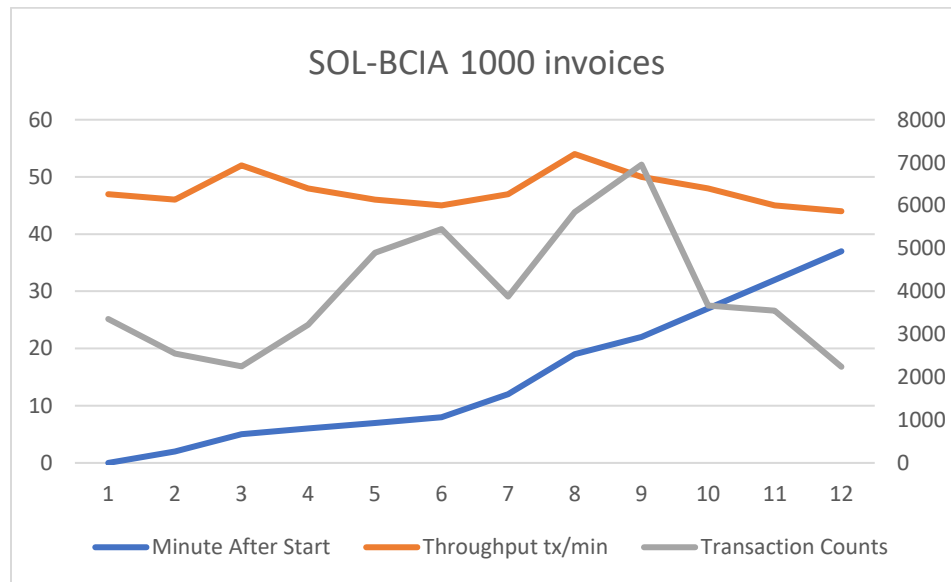


Figure 16 SOL-BCIA bot performance at 1000 invoices

Transaction fees and Finality Time: As shown in Table 20, Solana’s Finality times averaged 12 seconds at each experiment iteration, allowing transactions to be finalized quickly and to accept new upcoming transactions at high speed and with extremely low transaction fees compared to the ETH-BCIA bot. When converting the cryptocurrency into \$USD, the transaction fees were minuscule, with the largest iteration only costing \$1.78, which starkly contrasts the ETH-BCIA bot’s \$84.31.

Table 20 Total transaction fees and Finality Time averages for SOL-BCIA Bot

Number of processed invoices	Total transactions	Total Transaction fees	Transaction fees in \$USD	Finality Time (average)
50	1577	0.0003976	0.05	12 secs
100	2750	0.0005661	0.07	10 secs
500	6663	0.0016801	0.21	13 secs
1000	57587	0.01452115844	1.78	12 secs

4.7 Experiment 3: BNB-BCIA Bot

Small business simulation: Similarly to the SOL-BCIA bot, the throughput was exponentially higher than the ETH-BCIA Bot, and it took only three minutes to complete creating invoice payment evidence on the BC in a total of 1632 transactions

as shown in Table 21, and 2316 transactions for 100 invoices as shown in

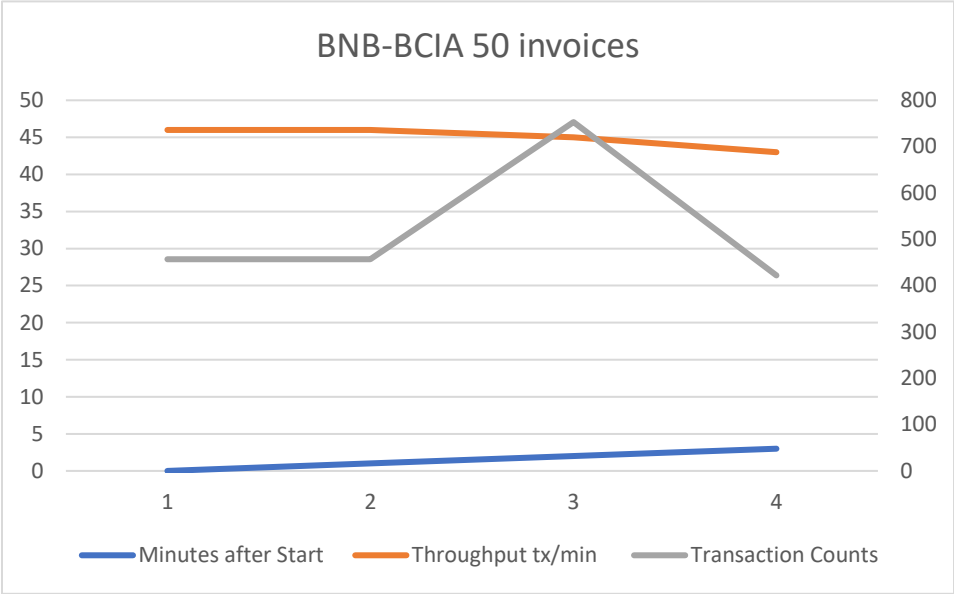


Figure 17 BNB-BCIA bot performance at 50 invoices

Table 22. As shown in Figure 17 and Figure 18, the transaction throughput remained consistent throughout the 50-invoice and 100-invoice iterations.

Table 21 BNB-BCIA Bot performance at 50 invoices

Minutes after Start	Throughput tx/min	Transaction Counts	Block Size
0	46	457	135677
2	45	753	174598
3	43	422	96322

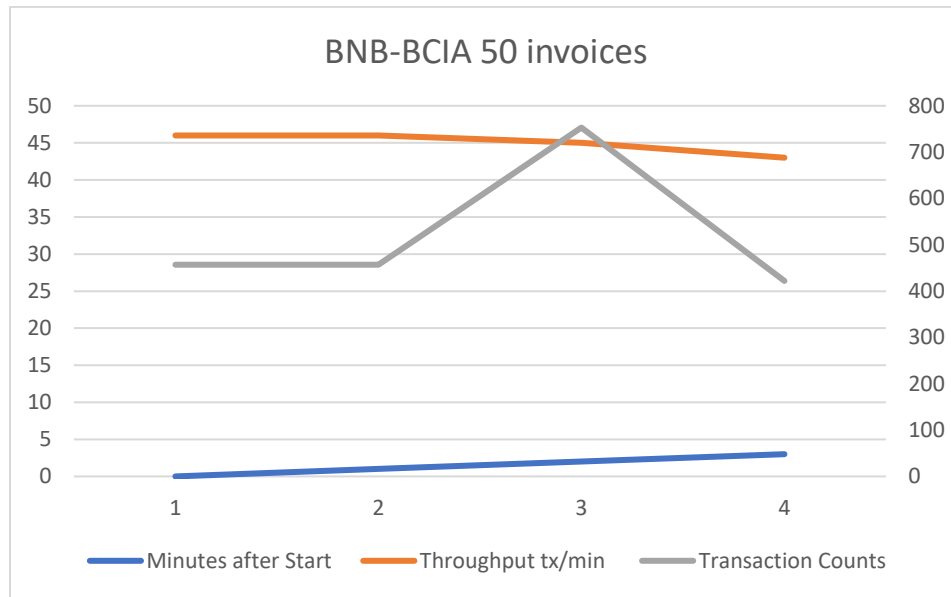


Figure 17 BNB-BCIA bot performance at 50 invoices

Table 22 BNB-BCIA Bot performance at 100 invoices

Minutes after Start	Throughput tx/min	Transaction Counts	Block Size
0	48	631	175841
1	46	753	192241
3	45	422	97051
5	47	510	134457

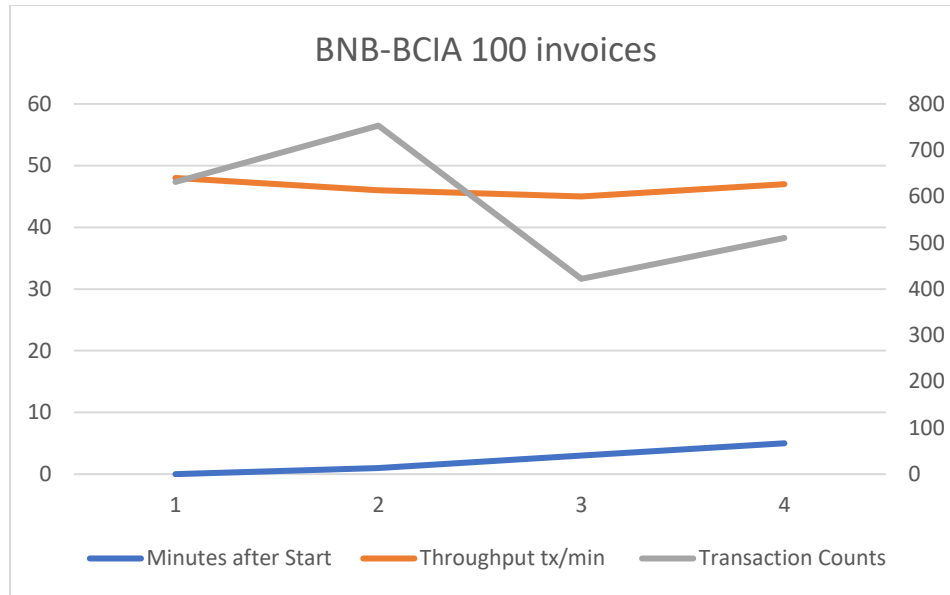


Figure 18 BNB-BCIA bot performance at 100 invoices

Medium Business simulation: At the 2nd minute of processing the invoice payment evidence, a spike in throughput and transactions was recorded as shown in Table 23, with block size varying from over 0.36 MB to a low of 0.1 MB bytes, where the maximum block size in BNB Chain is 1 MB. In Figure 19, there was a large dip from the 4th minute from 2542 transactions to 814 transaction then continued to lower gradually to 422 transactions bu the end of the iteration.

Table 23 BNB-BCIA Bot performance for 500 invoices

Minutes After Start	Throughput tx/min	Transaction Counts	Block Size
0	45	1187	195478
2	51	2542	367315
4	46	814	160207
7	48	651	108931
13	43	422	145263

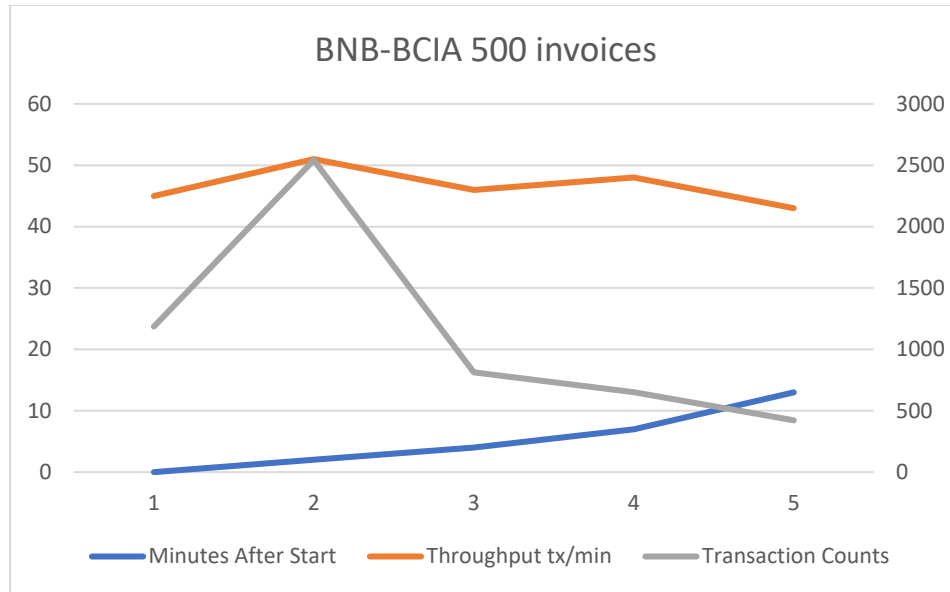


Figure 19 BNB-BCIA bot performance at 500 invoices

Large business simulation: Throughout the four iterations, the BNB-BCIA bot performed consistently with stable throughput and quick processing times compared to ETH-BCIA bot. As with the other bot versions at 1000 invoices, this iteration took the longest period of time, 32 minutes, as shown in Table 24 and in Figure 20, the transaction counts fluctuated from 3354 transactions to a high of 6954 transactions.

Table 24 BNB-BCIA Bot performance at 1000 invoices

Minute After Start	Throughput tx/min	Transaction Counts	Block Size
0	48	3354	119282
2	51	2547	159876
5	52	2254	129752
6	48	3221	143245
7	47	4899	145657
8	45	5448	123455
16	47	3877	114785
19	54	5849	358420
22	52	6954	289994

25	48	3669	157448
32	45	3548	185424
40	44	2241	186654

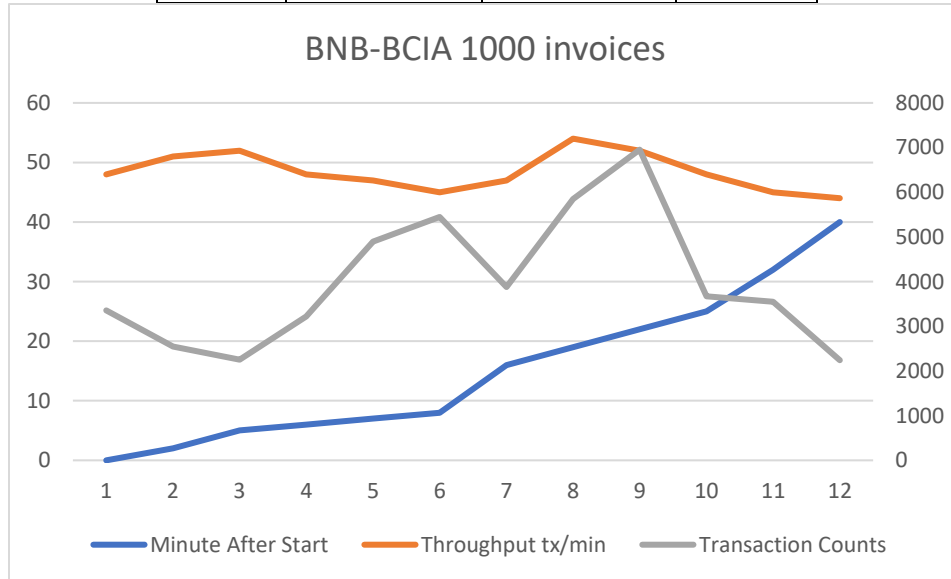


Figure 20 BNB-BCIA bot performance at 1000 invoices

Transaction fees and Finality Times: As shown in Table 25, Finality times averaged at around 8 secs except for the highest invoice processing iteration, where the finality time reached 11 secs. The Finality times of BNB-BCIA Bot were faster and superior than the SOL-BCIA bot and the ETH-BCIA bot, thus making BNB Chain a reliable BC network for DApps with competitive transaction fees, at only \$USD 4.20 for processing 1000 invoices.

Table 25 Transaction fees and Finality Times for BNB-BCIA Bot

Number of processed invoices	Total transactions	Total Transaction fees	Transaction fees in \$USD	Finality Time (average)
50	1577	0.0003976	0.17	8 secs
100	2750	0.0005661	0.33	7.7 secs
500	6663	0.0016801	1.84	9 secs
1000	57587	0.01452115844	4.20	11 secs

4.8 Cost Effectiveness

After completing all the experimentations at different transaction volumes across all three testnets, the researcher analyzed the cost-effectiveness of shifting the BCIA bot framework onto Mainnet networks, as testnets are only used for testing and debugging DApps and relying on testnets even for lightweight applications is ill-advised as testnets run out of tokens over time and deprecate like Goerli and Ropsten testnets in the past (see Table 2). Transaction fees differ from one Mainnet to another and the prices also fluctuate during the day, as illustrated in Figure 21 ; thus, for the purpose of this research, the weekly gas price average according to the Etherscan website's Ethereum Gas Tracker (etherscan.io, 2024), BNB Chain Explorer (BscScan.com, 2024), and the Solana Explorer (*Solana Fees + Burn Tracker - Why Are Solana's "Gas" Fees For Transactions So Low?*, 2024) websites was used, as shown in Table 26. As expected, ETH's gas fees were the most expensive out of the three cryptocurrencies during the week of 7-14 April 2024.

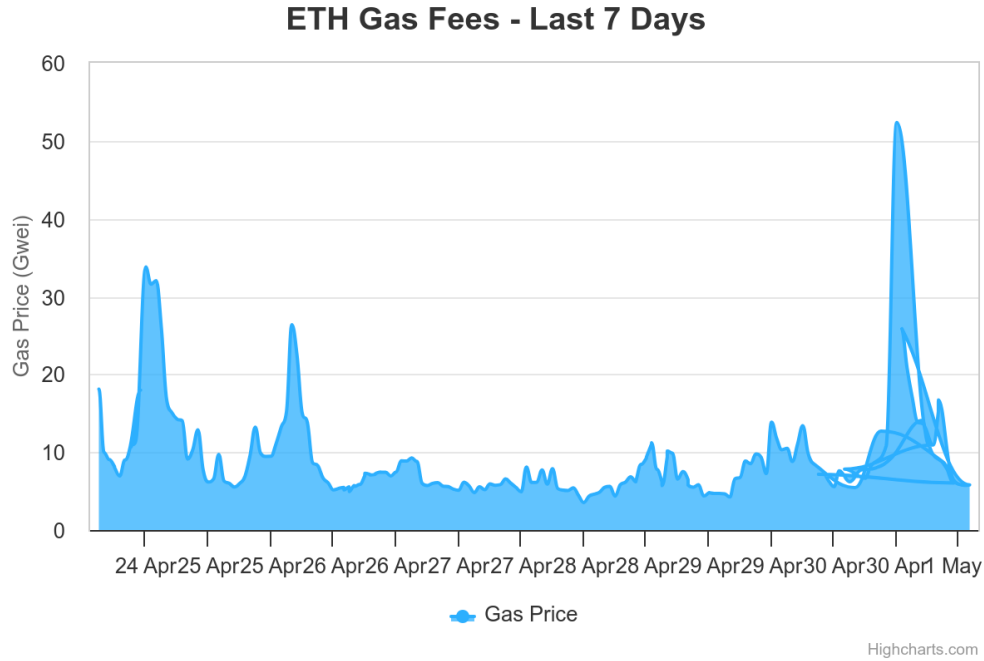


Figure 21 An example of Ethereum gas fees fluctuation over a 7-day period (Ethereum Gas Fees , 2024)

Table 26 Average Transaction fees in cryptocurrency and \$USD (7-14 April 2024)

	ETH	Solana	BNB Chain
Average transaction/ gas fees	18 Gwei	0.000100988 SOL	1 Gwei
Transaction fee/ gas price in \$USD	\$1.05	\$0.01	\$0.01

The following tables describe the costs in \$USD for every experiment iteration. Mainly due to the higher and fluctuating price of Ethereum gas, ETH-BCIA bot was by far the most costly of the three BCIA bots as shown in Table 27. As for Table 28, due to Solana's meager transaction fee starting from 0.000100988 SOL, which converts to \$USD 0.01, the transaction costs of SOL-BCIA Bot are a minuscule fraction of ETH-BCIA bot making the Solana version of the bot high performance and cost-effective. BNB-BCIA Bot can also be considered cost-effective as the highest iteration cost only \$USD 4.20, and also delivered high performance in all

four iterations without any noticeable deterioration in throughput, latency, nor finality time.

Table 27 ETH-BCIA bot's ETH transaction fees to \$USD conversion

Invoice Iteration	ETH	\$USD
50	0.0033993	10.12
100	0.0067268	20.03
500	0.013807	41.11
1000	0.02831665	84.31

Table 28 SOL-BCIA Bot's SOL transaction fee conversion to \$USD

Invoice Iteration	SOL	\$USD
50	0.0003976	0.05
100	0.0005661	0.07
500	0.0016801	0.21
1000	0.01452115844	1.78

Table 29 BNB-BCIA Bot transaction fee conversion to \$USD

Invoice Iteration	BNB	\$USD
50	0.00030411	0.17
100	0.000605	0.33
500	0.00337541	1.84
1000	0.00768791	4.20

4.9 Discussions

This chapter presented the experiment structure and results of testing the BCIA Bot Framework in three different versions to answer *RQ2: How do these challenges and opportunities impact small and medium-sized enterprises' overall efficiency, accuracy, and cost-effectiveness of invoice processing systems?* After completing all the experiment iterations for the three BCIA bot versions (ETH, SOL, BNB), it was found that all three versions successfully processed the invoices and generated the necessary activity logs, payment receipts, and BC evidence for all iterations at 50, 100, 500, and 1000 invoices without any system failures. The ETH-BCIA bot was the least cost-effective BCIA bot because transaction gas fees, at the time of experimenting, was high in comparison to the transaction fees of the SOL-BCIA bot and BNB-BCIA bot as shown in tables Table 27, Table 28, and Table 29.

When comparing the Block-DEF (Tian et al., 2019) and BCFL logging frameworks (Awuson-David et al., 2021) with the BCIA Bot Framework, the BCFL logging framework utilized a single-case mechanism experiment to validate the model (Wieringa, 2014) where the acquired log evidence was utilized in a digital forensic investigation. Similarly, the BCIA Bot Framework successfully generated activity logs onto the BC network and provided evidence of payment receipts. The BCFL logging framework utilized three performance metrics: Transaction rate, throughput, and latency. Table 30 presents the average findings of the throughput and latency from the BCFL literature and the BCIA Bot Framework experiments. The BCFL framework utilized a BFT consensus algorithm on a Hyperledger Fabric network, a private permissioned BC, gave high network throughput and lower latency results than the BCIA Bot Framework.

Table 30 Comparison of BCFL and BCIA Bot network throughput and latency

Framework	Averages	
	Network Throughput (tx/mn)	Latency (secs)
BCFL	~200	~4 secs
ETH-BCIA	~2.5	~16 secs
SOL-BCIA	~47	~8 secs
BNB-BCIA	~48	~6 secs

As for comparing the Block-DEF and BCIA Bot Framework performances as shown in Table , Block-DEF utilized five properties: scalability, integrity, validity, privacy, and traceability, and adopted either a quantitative or a qualitative analysis method. Regarding scalability, both Block-DEF and BCIA Bot frameworks achieved scalability as both frameworks were able to complete transactions utilizing different consensus algorithms (NPBFT and O-NPBFT) for Block-DEF and PoS, PoSA, and PoH for the BCIA Bot Framework and at multiple workloads. As for integrity and validity, both frameworks generated the necessary evidence on the BC ledger, which became immutable. Any manipulation or tampering of the original data can be detected easily by comparing the hash value of the evidence with that of the original file. The framework's validity is predominately based on the causal effect of integrity, which leads to the validity of using BC evidence and the loose coupling of data to ensure the validity of the BC evidence. As for the privacy property, BC technologies are built for privacy, and network participants' identities are sealed, thus making tracking difficult. In the case of Block-DEF, the traceability of participants is supported through a traceable key pair. The BCIA Bot Framework utilizes a different approach as it relies on the activity logs' metadata stored on the

BC to provide the traceability required to conduct any digital forensics investigations. Therefore, both Block-DEF and the BCIA Bot frameworks succeed in providing scalability, integrity, validity, privacy, and traceability.

Table 31 Comparison between Block-DEF and BCIA Framework

	Block-DEF	BCIA Framework
Scalability	Achieved scalability at multiple workloads	Achieved scalability through simulating invoice processing at 50, 100, 500, and 1000 invoices
Integrity	Successfully generated digital evidence on the BC ledger	Successfully generated digital evidence on the BC ledger
Validity	Utilized loose coupling of data to ensure the validity of the BC evidence	Storing the evidence on the BC ensured its validity
Privacy	Privacy is achieved as the evidence is stored on the BC	Privacy is achieved as the evidence is stored on the BC
Traceability	The traceability of participants is supported through a traceable key pair	The activity logs' metadata stored on the BC to provide the traceability required to conduct any digital forensics investigations

As for answering *RQ3: Which BC consensus algorithm provides optimal performance for BCIA bots for invoice processing?* The results of the experiment made the researcher believe that the BNB-BCIA bot was the best option overall with its utilization of BNB Chain. It brought high performance, with fast transaction throughputs, and could complete all iterations consistently and at low transaction

fees. Although the SOL-BCIA bot achieved similar high performances, Solana has experienced multiple service outages to their Mainnet and Devnet networks, with some lasting over 48 hours, thus inconveniencing users. Overall, based on the results of the experiments, SMEs can utilize the BCIA bot framework to process invoices and protect their systems from data leakage and digital fraud with minimal overhead to their businesses as the costs are operation-oriented and occur on a per-transaction basis (Fokaefs et al., 2021), where in contrast, creating a private blockchain network is a costly operation, with Ernest and Young estimating a yearly cost of \$USD 313,000 for large enterprises (*Total Cost of Ownership for Blockchain Solutions*, 2019)

Chapter 5 Conclusions and Recommendations

5.1 Conclusions

This chapter summarizes the overall research results of this thesis and the researcher's pursuit of answering its three primary research questions. In this thesis, the researcher presented their motivations and successfully attempted to develop a BC-based Intelligent Automation bot to combat digital fraud and data leakage utilizing BC, AI, and cloud computing technologies.

The first of the three research questions posed in this thesis *RQ1: What are the key security challenges and remedy solution opportunities associated with implementing IA bots powered by Blockchain in B2B e-commerce, particularly invoice processing?* Was answered through a systematic literature review aimed at finding the current security challenges in RPA and IA bots, in addition to introducing use cases of BC for invoice processing and its benefits and use cases of RPA and IA bots utilizing BC technologies for different sectors such as healthcare, education, and supply chain management amongst others. Eight key security challenges were synthesized from the systematic literature review, with logging and auditing being the highest-frequency security challenges. Researchers have a consensus that BC technologies provide an excellent opportunity to secure important data and make them immutable to ensure protection and integrity. Despite a lack of literature discussing the explicit utilization of BC-based IA bots in invoice processing, two BC secure-evidence frameworks were used to serve as a benchmark for the development of the BCIA Bot Framework.

As for *RQ2: How do these challenges and opportunities impact small and medium-sized enterprises' overall efficiency, accuracy, and cost-effectiveness of invoice*

processing systems? The Design Science Research Methodology (DSRM) was selected for its utility in creating and evaluating IT artifacts to solve a design problem: developing a BC-based Intelligent Automation bot to combat digital fraud and data leakage, known as the BCIA Bot Framework. In addition, both the Block-DEF and BCFL Logging frameworks utilized the DSRM methodology in developing their works, thus adding a solid foundation and justification for comparing all three frameworks and measuring their effectiveness and success in creating immutable digital evidence. Following the DSRM process model, two key RPA/ IA cybersecurity scenarios explain how cybercriminals can exploit RPA/IA bots to commit digital fraud by falsifying invoices and transferring funds elsewhere. As for data leakage, the scenario described how a cybercriminal would manipulate the data stored in the activity log or clear their tracks by deleting them. After providing the necessary context for the research, the BCIA Bot Framework is detailed and explained as a novel approach and system to combat these challenges. The BCIA Bot Framework utilizes a two-layer system, a Blockchain Layer and a Cloud Layer, to process invoices automatically through an IA bot with Optical Character Recognition capabilities and creates and stores activity logs and payment receipt evidence on a public BC network. Finally, the BCIA security architecture is discussed and compared with two other BC-based systems found in the literature, Block-Def and BCFL Logging framework, which aim to secure digital evidence for forensic investigations. To provide a quantitative analysis of the BCIA Bot Framework's performance, several experiments were designed and conducted to measure five performance metrics derived from the literature, which encompassed Network Throughput, Block size, Transaction fees, Transaction Latency, and Finality Time. The results proved that the BCIA Bot Framework provided an efficient, accurate, and cost-effective solution for SMEs to combat digital fraud and data leakage in IA invoice processing bots.

As for the final research question *RQ3: Which BC consensus algorithm provides optimal performance for BCIA bots for invoice processing?* The framework was tested on three BC testnets: Sepolia ETH, Solana Devnet, and BNB Chain, with each testnet running through four experiment iterations: Processing 50 and 100 invoices to simulate the workload of a typical small business, then a medium-sized business with a invoice processing workload of 500 invoices, and 1000 invoices to represent the workload of a large enterprise. The BCIA Bot showed variations in performance across all three testnets, where ETH-BCIA bot was able to complete all the iterations smoothly but it took longer time in comparison to SOL-BCIA Bot and BNB-BCIA bot. The latter two bots achieved both high performances and low transaction fees. However, after considering network stability over time, BNB Chain provided the best combination of cost-effectiveness and performance, making it particularly suitable for small and medium-sized enterprises.

In conclusion, this research presented the necessity of providing better security for RPA and IA activity logs by adding BC elements within an IA bot framework. This will provide additional security to detect and deter malicious actors from accessing bots aimed at committing digital fraud or causing data leakage. The BCIA Bot demonstrated effective, secure logging and auditing practices essential for RPA and IA applications. These insights validate the effectiveness of the BCIA Bot Framework and open pathways for further research into blockchain's role in securing automated business processes.

The study's main limitation was the lack of resources to implement the BCIA Bot Framework onto Mainnets, as testnets are not complete reflections of the capabilities of their respective networks, as many have throttles on some metrics such as throughput and block size. A threat to the experiment's validity is the fluctuation of

transaction fees throughout the day, which may affect the actual cost of utilizing the BCIA Bot Framework.

Moreover, we can deduce from the experiment results and comparison with Block-DEF and BCFL frameworks that utilizing public BC networks such as Ethereum, Solana, and BNB chain does have drawbacks as their networks are dynamic and rely on incentives to process transactions, bringing with them the uncertainties of fluctuating transaction fees and gas prices which can lead to unexpected increases in overhead in SMEs, particularly when using the Ethereum BC as it is considered to be the most trusted BC network at the moment, even though it suffers from slower finality speed and high transaction fees. Another drawback is the increasing possibility of service outages as user numbers grow, leading to network congestion due to the high amounts of transactions. Solana, in particular, has experienced multiple outages on their Devnet and Mainnet BC networks in the last two years since its launch. For example, the most prolonged outage lasted almost 48 hours in February 2023 when a bottleneck within the “Turbine” primary block-propagation protocol saturated its capacity and led to the network failing (*02-25-23 Solana Mainnet Beta Outage Report*, 2023). A year later, in February 2024, a five-hour outage was caused by a software bug in a translator function that created a domino effect, halting block production (Chanda, 2024). At the time of writing, Solana experienced another outage in its Devnet network in April 2024, but the reasons for the outage have not yet been made official (Patel, 2024).

However, the experiment also raises more questions on how each network handles creating blocks to process the BCIA bot’s requests, as the block sizes and transaction fees differ completely at each experiment iteration with no clear patterns to draw from and understand. The researcher believes this brings a great opportunity for researchers and practitioners to devise and develop a comprehensive framework to

determine which Public BC platform works best for individual dApps, including performance metrics related to the dApp itself and not the network situation as no individual has control over of the BC network. Examples of possible dApp performance criteria could be scalability, execution time, and operation latency.

the motivations of using a permissioned blockchain become more logical and evident as the developers have total control over the BC architecture and utilize it to their own advantage and manage it, unlike using a Public permissionless BC. However, this does not diminish the fact that the BCIA Bot framework successfully simulated a small business processing invoices and protecting their IA bot from malicious actors attempting to commit digital fraud or data leakage to a large enterprise processing thousands of invoices daily.

5.2 Future Work and Recommendations

The results of the experiments discussed in the thesis shed light on the necessity of developing a robust testing and performance monitoring framework for BC- based Intelligent Automation bots and for DApps in general, and in particular, DApps looking to utilize public permissionless BCs as a platform. As seen by the results of Chapter Four, public BCs performances change daily and do not provide a consistent and decisive result, as many different variables are at play. Researchers and practitioners would benefit immensely from such a framework to help inform them which pathway to choose when developing DApps. Also, as an extension to this body of work, the researcher suggests that the BCIA Bot Framework be implemented onto BC Mainnets to get a better insight into the true performance of the BCIA Bot and its cost-effectiveness for SMEs. In addition, the researcher believes that further research is required in developing BaaS services to cater to SMEs to enhance

security and automation, which are areas that guarantee thriving economic growth as SMEs are the backbone of any developed country's economy.

Building BC-based Intelligent Automation bots remains a nascent field of research, and in the current cybersecurity landscape, it is constantly changing and shifting toward new and unprecedented cyber threats. Quantum computing is the biggest challenge facing cybersecurity experts as its immense processing power can dismantle BC's traditional encryption methods RSA and ECC with potentially great ease, leaving sensitive personal data stored on BCs across the globe vulnerable to data theft and manipulation.

Appendix

One conference paper was derived from this research. Also, two other conference papers and a journal article were published as part of the Computing and Information Sciences PhD program. The papers are attached in the subsequent pages.

References:

- 02-25-23 Solana Mainnet Beta Outage Report. (2023, June 24). Solana.
<https://solana.com/news/02-25-23-solana-mainnet-beta-outage-report>
- Alexopoulos, C., Charalabidis, Y., Androutsopoulou, A., Loutsaris, M. A., & Lachana, Z. (2019). *Benefits and Obstacles of Blockchain Applications in e-Government*. <https://hdl.handle.net/10125/59773>
- Al-Slais, Y. (2020). Privacy Engineering Methodologies: A survey. 2020 *International Conference on Innovation and Intelligence for Informatics, Computing and Technologies, 3ICT 2020*.
<https://doi.org/10.1109/3ICT51146.2020.9311949>
- Al-Slais, Y., & Ali, M. (2023). Robotic Process Automation and Intelligent Automation Security Challenges: A Review. 2023 *International Conference On Cyber Management And Engineering (CyMaEn)*, 71–77.
<https://doi.org/10.1109/CyMaEn57228.2023.10050996>
- Al-Slais, Y., & El-Medany, W. (2022). User-Centric Adaptive Password Policies to Combat Password Fatigue. *International Arab Journal of Information Technology*, 19(1). <https://doi.org/10.34028/iajit/19/1/7>
- Alzoubi, Y. I., & Mishra, A. (2024). Techniques to alleviate blockchain bloat: Potentials, challenges, and recommendations. *Computers and Electrical Engineering*, 116, 109216.
<https://doi.org/10.1016/j.compeleceng.2024.109216>
- Amo, D., Fonseca, D., Alier, M., García-Peñalvo, F. J., Casañ, M. J., & Alsina, M. (2019). Personal Data Broker: A Solution to Assure Data Privacy in EdTech.

- Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 11590 LNCS, 3–14. https://doi.org/10.1007/978-3-030-21814-0_1
- Ampel, B. M. (2023). *The Effect of Consensus Algorithm on Ethereum Price and Volume*.
- Andriole, S. J. (2020). Blockchain, Cryptocurrency, and Cybersecurity. *IT Professional*, 22(1), 13–16. <https://doi.org/10.1109/MITP.2019.2949165>
- ASIC-Resistant. (2023, July 4). <https://www.rain.com/en-bh/glossary/asic-resistant>
- Austin, T. H., & Di Troia, F. (2022). A Blockchain-Based Tamper-Resistant Logging Framework. *Communications in Computer and Information Science*, 1683 CCIS, 90–104. https://doi.org/10.1007/978-3-031-24049-2_6
- Australian National University, Australia, Gregor, S., Kruse, L., University of Liechtentsein, Liechtentsein, Seidel, S., & University of Liechtentsein, Liechtentsein. (2020). Research Perspectives: The Anatomy of a Design Principle. *Journal of the Association for Information Systems*, 21, 1622–1652. <https://doi.org/10.17705/1jais.00649>
- Awuson-David, K., Al-Hadhrani, T., Alazab, M., Shah, N., & Shalaginov, A. (2021). BCFL logging: An approach to acquire and preserve admissible digital forensics evidence in cloud ecosystem. *Future Generation Computer Systems*, 122, 1–13. <https://doi.org/10.1016/j.future.2021.03.001>
- Bamakan, S. M. H., Motavali, A., & Babaei Bondarti, A. (2020). A survey of blockchain consensus algorithms performance evaluation criteria. *Expert Systems with Applications*, 154, 113385. <https://doi.org/10.1016/j.eswa.2020.113385>
- Basile, D., Goretti, V., Di Ciccio, C., & Kirrane, S. (2021). Enhancing Blockchain-Based Processes with Decentralized Oracles. *Lecture Notes in Business*

- Information Processing*, 428, 102–118. https://doi.org/10.1007/978-3-030-85867-4_8/TABLES/3
- Battaiola, E., Massacci, F., Ngo, C. N., & Sterlini, P. (2019). *Blockchain-based Invoice Factoring: From business requirements to commitments*.
- Baviskar, D., Ahirrao, S., Potdar, V., & Kotecha, K. (2021). Efficient Automated Processing of the Unstructured Documents Using Artificial Intelligence: A Systematic Literature Review and Future Directions. *IEEE Access*, 9, 72894–72936. <https://doi.org/10.1109/ACCESS.2021.3072900>
- Beck, P., Bockrath, H., Knoche, T., Digtiar, M., Petrich, T., Romanchenko, D., Hobeck, R., Pufahl, L., Klinkmüller, C., & Weber, I. (2021). *BLF: A Blockchain Logging Framework for Mining Blockchain Data*. <http://ceur-ws.org>
- Becoming Strategic with Intelligent Automation*. (2020). <https://doi.org/10.17705/2msqe.000XX>
- Bhamidipati, N. R., Vakkavanthula, V., Stafford, G., Dahir, M., Neupane, R., Bonnah, E., Wang, S., Murthy, J. V. R., Hoque, K. A., & Calyam, P. (2021). ClaimChain: Secure Blockchain Platform for Handling Insurance Claims Processing. *2021 IEEE International Conference on Blockchain (Blockchain)*, 55–64. <https://doi.org/10.1109/Blockchain53845.2021.00019>
- Biswas, A., & Dutta, P. K. (2021). Novel Approach of Automation to Risk Management: The Reduction in Human Errors. *EAI/Springer Innovations in Communication and Computing*, 683–696. https://doi.org/10.1007/978-3-030-49795-8_65
- Brocke, J. vom, Hevner, A., & Maedche, A. (2020). *Introduction to Design Science Research* (pp. 1–13). https://doi.org/10.1007/978-3-030-46781-4_1
- BscScan.com. (2024). *BNB Smart Chain (BNB) Blockchain Explorer*. BNB Smart Chain Explorer. <https://bscscan.com/>

- Carter, R. (2023, August 21). Gartner RPA Magic Quadrant 2023: The Latest Updates. *CX Today*. <https://www.cxtoday.com/data-analytics/gartner-magic-quadrant-for-robotic-process-automation-rpa-2022/>
- Catt, M. (2019, March 31). Blockchain Fundamentals: Latency & Capacity — Featuring the Ark Ecosystem. *University of Kansas Blockchain Institute*. <https://medium.com/ku-blockchain-institute/blockchain-fundamentals-featuring-the-ark-ecosystem-part-1-af1f9052e579>
- Chainspect. (2024, January 25). *What is Time to Finality (TTF)? [Real TTF in 2024]*. Chainspect. <https://chainspect.app/blog/time-to-finality-ttf>
- Chakraborti, T., Isahagian, V., Khalaf, R., Khazaeni, Y., Muthusamy, V., Rizk, Y., & Unuvar, M. (2020). From Robotic Process Automation to Intelligent Process Automation: – Emerging Trends –. *Lecture Notes in Business Information Processing*, 393 LNBIP, 215–228. https://doi.org/10.1007/978-3-030-58779-6_15
- Chanda, P. (2024, February 12). Despite Solana’s recent outage, this exec remains positive. *AMBCrypto*. <https://ambcrypto.com/despite-solanas-recent-outage-this-executive-remains-optimistic/>
- Chaudhuri, A., Bhatia, M. S., Subramanian, N., Kayikci, Y., & Dora, M. (2022). Socio-technical capabilities for blockchain implementation by service providers: Multiple case study of projects with transaction time reduction and quality improvement objectives. *Production Planning & Control*, 0(0), 1–14. <https://doi.org/10.1080/09537287.2022.2128865>
- Chen, L., Lou, F., & Fu, T. (2022). Automatic Analysis of RPA in Digital Service Testing of Grid System. *Lecture Notes on Data Engineering and Communications Technologies*, 98, 858–863. https://doi.org/10.1007/978-3-030-89511-2_117/FIGURES/1

- Choi, D., R'bigui, H., & Cho, C. (2021). Robotic Process Automation Implementation Challenges. *Lecture Notes in Networks and Systems*, 149, 297–304. https://doi.org/10.1007/978-981-15-7990-5_29
- Comparing Blockchain Fee Structures: Comprehensive Guide. (2023). Tatum. <https://tatum.io/blog/blockchain-fees-comparison-guide>
- Consensus Algorithms in Blockchain. (2019, April 25). *GeeksforGeeks*. <https://www.geeksforgeeks.org/consensus-algorithms-in-blockchain/>
- Costa, D. A. da S., Mamede, H. S., & Mira da Silva, M. (2022). Robotic Process Automation (RPA) Adoption: A Systematic Literature Review. *Engineering Management in Production and Services*, 14(2), 1–12. <https://doi.org/10.2478/emj-2022-0012>
- Courtois, N. T., Emirdag, P., & Valsorda, F. (2014). *Private Key Recovery Combination Attacks: On Extreme Fragility of Popular Bitcoin Key Management, Wallet and Cold Storage Solutions in Presence of Poor RNG Events*.
- Dahlia Fernandez & Aini Aman. (2021). The Challenges of Implementing Robotic Process Automation in Global Business Services. *International Journal of Business and Society*, 22(3), 1269–1282. <https://doi.org/10.33736/ijbs.4301.2021>
- Desai, D., Jain, A., Naik, D., Panchal, N., & Sawant, D. (2021). Invoice Processing using RPA & AI. *SSRN Electronic Journal*. <https://doi.org/10.2139/ssrn.3852575>
- Eggers, J., Hein, A., Weking, J., Böhm, M., & Krcmar, H. (2021). Process automation on the blockchain: An exploratory case study on smart contracts. *Proceedings of the Annual Hawaii International Conference on System Sciences*, 2020-Janua, 5607–5616. <https://doi.org/10.24251/hicss.2021.681>

- Enriquez, J. G., Jimenez-Ramirez, A., Dominguez-Mayo, F. J., & Garcia-Garcia, J. A. (2020). Robotic Process Automation: A Scientific and Industrial Systematic Mapping Study. *IEEE Access*, 8, 39113–39129.
<https://doi.org/10.1109/ACCESS.2020.2974934>
- Ethereum Gas Fees*. (2024). Milk Road. <https://milkroad.com/ethereum/gas/>
- Etherscan.io*. (2024). *Ethereum Gas Tracker* | *Etherscan*. Ethereum (ETH) Blockchain Explorer. <https://etherscan.io/gastracker>
- Ferdous, M. S., Chowdhury, M. J. M., Hoque, M. A., & Colman, A. (2020). *Blockchain Consensus Algorithms: A Survey* (arXiv:2001.07091). arXiv.
<http://arxiv.org/abs/2001.07091>
- Filippi, P. D., Wray, C., & Sileno, G. (2021). Smart contracts. *Internet Policy Review*, 10(2). <https://doi.org/10.14763/2021.2.1549>
- Flechsigg, C., Anslinger, F., & Lasch, R. (2022). Robotic Process Automation in purchasing and supply management: A multiple case study on potentials, barriers, and implementation. *Journal of Purchasing and Supply Management*, 28(1), 100718.
<https://doi.org/10.1016/J.PURSUP.2021.100718>
- Fokaefs, M., Rasolroveicy, M., & Haouari, W. (2021). *Public or Private? A Techno-Economic Analysis of Blockchain*.
- Forment, M. A., Filvà, D. A., García-Peñalvo, F. J., Escudero, D. F., & Casañ, M. J. (2018). Learning analytics' privacy on the blockchain. *ACM International Conference Proceeding Series*, 294–298.
<https://doi.org/10.1145/3284179.3284231>
- Four Steps to Ensure Robotic Process Automation Security* | *Gartner*. (2022).
<https://www.gartner.com/en/doc/735800-four-steps-to-ensure-robotic-process-automation-security>

- Gartner, G. (2023). *Best Robotic Process Automation Reviews 2023* | Gartner Peer Insights. Gartner. <https://www.gartner.com/market/robotic-process-automation>
- Giesbers, S. (2020). Robotic process automation and internal control: A guideline. *Research in IT-Auditing: A Multidisciplinary View*, 171.
- Gotthardt, M., Koivulaakso, D., Paksoy, O., Saramo, C., Martikainen, M., & Lehner, O. (2020). ACRN Journal of Finance and Risk Perspectives Current State and Challenges in the Implementation of Smart Robotic Process Automation in Accounting and Auditing. *ACRN Journal of Finance and Risk Perspectives*, 9, 90–102. <https://doi.org/10.35944/jofrp.2020.9.1.007>
- Griffiths, L., & Pretorius, H. W. (2021). Implementing Robotic Process Automation for Auditing and Fraud Control. *Communications in Computer and Information Science*, 1477 CCIS, 26–36. https://doi.org/10.1007/978-3-030-86761-4_3/FIGURES/1
- Guarda, T., Bustos, S., Espada, M. C., & Garallar, D. G. (2024). Blockchain and Robotic Process Automation Working Together. In T. Guarda, F. Portela, & J. M. Diaz-Nafria (Eds.), *Advanced Research in Technologies, Information, Innovation and Sustainability* (Vol. 1935, pp. 521–530). Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-48858-0_41
- Guerar, M., Merlo, A., Migliardi, M., Palmieri, F., & Verderame, L. (2020). A Fraud-Resilient Blockchain-Based Solution for Invoice Financing. *IEEE Transactions on Engineering Management*, 67(4), 1086–1098. <https://doi.org/10.1109/TEM.2020.2971865>
- Guidehouse. (2021). *RPA Security*. <https://guidehouse.com/insights/advanced-solutions/2021/rpa-security>

- Guo, H., & Yu, X. (2022). A survey on blockchain technology and its security. *Blockchain: Research and Applications*, 3(2), 100067.
<https://doi.org/10.1016/J.BCRA.2022.100067>
- Haddara, M., & Zach, O. (2011). ERP Systems in SMEs: A Literature Review. *2011 44th Hawaii International Conference on System Sciences*, 1–10.
<https://doi.org/10.1109/HICSS.2011.191>
- Hameed, K., Barika, M., Garg, S., Amin, M. B., & Kang, B. (2022). A taxonomy study on securing Blockchain-based Industrial applications: An overview, application perspectives, requirements, attacks, countermeasures, and open issues. *Journal of Industrial Information Integration*, 26.
<https://doi.org/10.1016/J.JII.2021.100312>
- Hartley, J. L., & Sawaya, W. J. (2019). Tortoise, not the hare: Digital transformation of supply chain business processes. *Business Horizons*, 62(6), 707–715. <https://doi.org/10.1016/J.BUSHOR.2019.07.006>
- Hattab, S., & Taha Alyaseen, I. F. (2019). Consensus Algorithms Blockchain: A comparative study. *International Journal on Perceptive and Cognitive Computing*, 5(2), 66–71. <https://doi.org/10.31436/ijpcc.v5i2.103>
- Hill, B., Chopra, S., Valencourt, P., & Prusty, N. (2018). *Blockchain Developer's Guide: Develop smart applications with Blockchain technologies - Ethereum, JavaScript, Hyperledger Fabric, and Corda*. Packt Publishing Ltd.
- Hong, B. (Yue), Ly, M., & Lin, H. (2023). Robotic Process Automation Risk Management: Points to Consider. *Journal of Emerging Technologies in Accounting*, 20(1), 125–145. <https://doi.org/10.2308/JETA-2022-004>
- How Much Does It Really Cost to Process an Invoice?* (2022, August 24). [Text]. AccountingWEB. <https://www.accountingweb.co.uk/community/industry-insights/how-much-does-it-really-cost-to-process-an-invoice>

- Huang, D., Ma, X., & Zhang, S. (2020). Performance Analysis of the Raft Consensus Algorithm for Private Blockchains. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 50(1), 172–181.
<https://doi.org/10.1109/TSMC.2019.2895471>
- IBM Blog, E. (2021, October 15). Differentiating Between Intelligent Automation and Hyperautomation. *IBM Blog*. <https://www.ibm.com/blog/differentiating-between-intelligent-automation-and-hyperautomation/>
- Introduction | BNB Chain Documentation. (2024).
<https://docs.bnbchain.org/docs/learn/intro/>
- Iqbal, M., & Matulevičius, R. (2019). *Comparison of Blockchain-Based Solutions to Mitigate Data Tampering Security Risk* (pp. 13–28).
https://doi.org/10.1007/978-3-030-30429-4_2
- Ivančić, L., Vugec, D. S., & Vukšić, V. B. (2019). Robotic Process Automation: Systematic Literature Review. *Lecture Notes in Business Information Processing*, 361, 280–295. https://doi.org/10.1007/978-3-030-30429-4_19
- Jurmu, P. (2019). *Towards Secure Robot Process Automation Environments*.
- Kaplan, J., Richter, W., & Ware, D. (2019). *Cybersecurity: Linchpin of the digital enterprise*.
- Kerutis, V., & Calneryte, D. (2022a). Intelligent Invoice Documents Processing Employing RPA Technologies. *Communications in Computer and Information Science*, 1665 CCIS, 235–247. https://doi.org/10.1007/978-3-031-16302-9_18
- Kerutis, V., & Calneryte, D. (2022b). *Intelligent Invoice Documents Processing Employing RPA Technologies*. 235–247. https://doi.org/10.1007/978-3-031-16302-9_18/FIGURES/9
- Kitchenbaum, B., & Charters, S. (2007). Guidelines for performing Systematic Literature Reviews in Software Engineering. *EBSE Technical Report*.

- Klaytn. (2022, July 20). A comparison of blockchain network latencies. *Klaytn*.
<https://medium.com/klaytn/a-comparison-of-blockchain-network-latencies-7508509b8460>
- Kosi, F. (2019). *Mercy College Robotic Process Automation (RPA) and Security A Thesis*.
- Kumzy. (2024, March 12). Unveiling the Power of Subspace: A Comparative Analysis of Proof of Capacity Against PoW and PoS. *Medium*.
<https://medium.com/@Upnode/unveiling-the-power-of-subspace-a-comparative-analysis-of-proof-of-capacity-against-pow-and-pos-a52c58a68523>
- Kurian, E., & Varghese, S. (2021). Relevance of Bots in Software and Their Impacts on Software Security. *Lecture Notes in Engineering and Computer Science*, 2242, 207–212.
- Kurylets, A., & Goranin, N. (2023). Security Ontology OntoSecRPA for Robotic Process Automation Domain. *Applied Sciences*, 13(9), Article 9.
<https://doi.org/10.3390/app13095568>
- La Rosa, M., Loos, P., & Pastor, O. (Eds.). (2016). *Business Process Management* (Vol. 9850). Springer International Publishing. <https://doi.org/10.1007/978-3-319-45348-4>
- Liu, X. (2018). Research and Application of Electronic Invoice Based on Blockchain. *MATEC Web of Conferences*, 232, 04012.
<https://doi.org/10.1051/matecconf/201823204012>
- Lok, C. P. (2021). *Critical Success Factors for Robotic Process Automation Implementation*. <http://hdl.handle.net/10292/14380>
- Low Code/No Code and RPA: Rewards and Risks*. (2023). Nodok Security.
<https://nokodsecurity.com/wp-content/uploads/2023/12/Low-CodeNo-Code-and-RPA-Rewards-and-Risks-Nokod->

Security_web.pdf?utm_medium=email&_hsenc=p2ANqtz-8aWi1LsbSMQ3G_9oq89lhCLm1kixhnzPeayhM7i-HsEl2LRhxnFbU-qviPAccGrrdYP9I_U0K0GzGfcKWSAJwPruc1lg&_hsmi=78627489&utm_content=78627489&utm_source=hs_automation

Madakam, S., Holmukhe, R. M., & Jaiswal, D. K. (2019). The Future Digital Work Force: Robotic Process Automation (RPA). *JISTEM - Journal of Information Systems and Technology Management*, 16, e201916001.

<https://doi.org/10.4301/S1807-1775201916001>

Man, J. (2022). *Towards the Future of Work: Managing the Risks of AI and Automation*. <https://doi.org/10.13140/RG.2.2.10084.45446>

Marciniak, P., & Stanisławski, R. (2021). Internal Determinants in the Field of RPA Technology Implementation on the Example of Selected Companies in the Context of Industry 4.0 Assumptions. *Information*, 12(6), 222.

<https://doi.org/10.3390/info12060222>

Mingxiao, D., Xiaofeng, M., Zhe, Z., Xiangwei, W., & Qijun, C. (2017). A review on consensus algorithm of blockchain. *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2567–2572.

<https://doi.org/10.1109/SMC.2017.8123011>

Moffitt, K. C., Rozario, A. M., & Vasarhelyi, M. A. (2018). Robotic Process Automation for Auditing. *Journal of Emerging Technologies in Accounting*, 15(1), 1–10. <https://doi.org/10.2308/JETA-10589>

Money, W. H. (2021). *Resolving Pressure and Stress on Governance Models from Robotic Process Automation Technologies*.

<https://proc.conisar.org>; <https://iscap.info>

Müller, M., Ostern, N., & Rosemann, M. (2020). Silver Bullet for All Trust Issues? Blockchain-Based Trust Patterns for Collaborative Business Processes.

- Lecture Notes in Business Information Processing*, 393 LNBIP, 3–18.
https://doi.org/10.1007/978-3-030-58779-6_1/TABLES/8
- Murugappan, K., & Sree Kala, T. (2022). *An Enhanced Security Framework for Robotic Process Automation* (pp. 231–238). Springer, Singapore.
https://doi.org/10.1007/978-981-16-3961-6_20
- Nakamoto, S. (2008). *Bitcoin: A Peer-to-Peer Electronic Cash System*.
- Nallasivam, A. (2022, January 28). Boosting RPA Security: Concerns, Solutions, and Best Practices. *ClaySys Technologies*.
<https://www.claysys.com/blog/rpa-security-practices/>
- Narayanam, K., Goel, S., Singh, A., Shrinivasan, Y., Chakraborty, S., Selvam, P., Choudhary, V., & Verma, M. (2020). Blockchain Based e-Invoicing Platform for Global Trade. *2020 IEEE International Conference on Blockchain (Blockchain)*, 385–392.
<https://doi.org/10.1109/Blockchain50366.2020.00056>
- Ng, K. K. H., Chen, C. H., Lee, C. K. M., Jiao, J. (Roger), & Yang, Z. X. (2021). A systematic literature review on intelligent automation: Aligning concepts from theory, practice, and future perspectives. *Advanced Engineering Informatics*, 47, 101246. <https://doi.org/10.1016/J.AEI.2021.101246>
- Nguyen, C. T., Hoang, D. T., Nguyen, D. N., Niyato, D., Nguyen, H. T., & Dutkiewicz, E. (2019). Proof-of-Stake Consensus Mechanisms for Future Blockchain Networks: Fundamentals, Applications and Opportunities. *IEEE Access*, 7, 85727–85745. <https://doi.org/10.1109/ACCESS.2019.2925010>
- Noppen, P., & Jonker, D. M. (2019). *The Qualitative Impact of Robotic Process Automation*.
- Ongaro, D., & Ousterhout, J. (2014). *In Search of an Understandable Consensus Algorithm*.

- Pandey, D. P., & Pandey, D. M. M. (2015). *RESEARCH METHODOLOGY: TOOLS AND TECHNIQUES*.
- Parasuraman, R., Sheridan, T. B., & Wickens, C. D. (2000). A model for types and levels of human interaction with automation. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, 30(3), 286–297.
<https://doi.org/10.1109/3468.844354>
- Patel, H. (2024, April 19). This “risky” stablecoin edges Tether, Circle, but... - Arthur Hayes. *AMBCrypto*. <https://ambcrypto.com/ethenas-risky-usde-has-an-edge-over-tether-circle-but-arthur-hayes/>
- Peppers, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). A Design Science Research Methodology for Information Systems Research. *Journal of Management Information Systems*, 24(3), 45–77.
<https://doi.org/10.2753/MIS0742-1222240302>
- Petersen, K., Vakkalanka, S., & Kuzniarz, L. (2015). Guidelines for conducting systematic mapping studies in software engineering: An update. *Information and Software Technology*, 64, 1–18.
<https://doi.org/10.1016/J.INFSOF.2015.03.007>
- Podgorelec, B., Keršič, V., & Turkanović, M. (2019). Analysis of Fault Tolerance in Permissioned Blockchain Networks. *2019 XXVII International Conference on Information, Communication and Automation Technologies (ICAT)*, 1–6. <https://doi.org/10.1109/ICAT47117.2019.8938836>
- Practical Byzantine Fault Tolerance(pBFT). (2019, January 11). *GeeksforGeeks*.
<https://www.geeksforgeeks.org/practical-byzantine-fault-tolerancepbft/>
- Proof of Capacity (PoC) in Blockchain—Shiksha Online*. (2023).
<https://www.shiksha.com/online-courses/articles/proof-of-capacity-in-blockchain/>

Proof of Importance. (2023, June 27). *Techopedia*.

<https://www.techopedia.com/definition/33599/proof-of-importance-poi>

Proof of Stake Authority (PoSA) Definition | CoinMarketCap. (2023).

CoinMarketCap Academy.

<https://coinmarketcap.com/alexandria/glossary/proof-of-stake-authority-posa>

Putz, B., Menges, F., & Pernul, G. (2019). *A secure and auditable logging infrastructure based on a permissioned blockchain*.

Ravi, V., & Kamaruddin, S. (2017). *Big Data Analytics Enabled Smart Financial Services: Opportunities and Challenges* (pp. 15–39).

https://doi.org/10.1007/978-3-319-72413-3_2

Raza, H., Baptista, J., & Constantinides, P. (2019). Conceptualizing the role of IS security compliance in projects of digital transformation: Tensions and shifts between prevention and response modes. *40th International Conference on Information Systems, ICIS 2019*.

Reducing RPA security risk | Security Magazine. (2023).

<https://www.securitymagazine.com/articles/98953-reducing-rpa-security-risk>

Robotic Process Automation Security and Why It's Important. (2022).

<https://www.impactmybiz.com/blog/robotic-process-automation-security/>

Rybinski, F., & Schöler, S. (2022). *Process Discovery Analysis for Generating RPA Flowcharts*. 231–245. https://doi.org/10.1007/978-3-031-16168-1_15

Sahu, S., Salwekar, S., Pandit, A., & Patil, M. (2020). Invoice Processing Using Robotic Process Automation. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, 216–223.

<https://doi.org/10.32628/CSEIT2062106>

Sanchez, H. L., Tysebaert, S., Rath, A., & Rivière, E. (2022). AuditTrust: Blockchain-Based Audit Trail for Sharing Data in a Distributed

- Environment. *Communications in Computer and Information Science*, 1656 CCIS, 5–17. https://doi.org/10.1007/978-3-031-16245-9_1
- Security, H. N. (2022, November 17). Robotic Process Automation (RPA) adoption hindered by security concerns. *Help Net Security*.
<https://www.helpnetsecurity.com/2022/11/17/robotic-process-automation-rpa-security-video/>
- Security Risks in Robotic Process Automation (RPA): How You Can Prevent Them* – CiGen. (2022). <https://www.cigen.com.au/security-risks-robotic-process-automation-rpa-how-prevent-them/>
- Shekhtman, L., & Waisbard, E. (2021). Engravechain: A Blockchain-based tamper-proof distributed log system. *Future Internet*, 13(6).
<https://doi.org/10.3390/fi13060143>
- Shrivastava, H. (2022, October 28). What Is Proof-of-Stake Authority (PoSA)? Are PoS And PoA Related To It? *WazirX Blog*. <https://wazirx.com/blog/what-is-proof-of-stake-authority-posa-are-pos-and-poa-related-to-it/>
- Sibanyoni, N. A. (2021). *A Blockchain-Based Robotic Process Automation Mechanism in Educational Setting* (pp. 17–41). <https://doi.org/10.4018/978-1-7998-6650-3.ch002>
- Singh, N., & Vardhan, M. (2020). Computing Optimal Block Size for Blockchain based Applications with Contradictory Objectives. *Procedia Computer Science*, 171, 1389–1398. <https://doi.org/10.1016/j.procs.2020.04.149>
- Smith, S. S. (2020). International Journal of Economics and Financial Issues Emerging Technologies and Implications for Financial Cybersecurity. *International Journal of Economics and Financial Issues* |, 10(1), 27–32.
<https://doi.org/10.32479/ijefi.8844>
- Sobreira Leite, G., Bessa Albuquerque, A., & Rogerio Pinheiro, P. (2020). Process Automation and Blockchain in Intelligence and Investigation Units: An

- Approach. *Applied Sciences*, 10(11), Article 11.
<https://doi.org/10.3390/app10113677>
- Solana Fees + Burn Tracker—Why Are Solana’s “Gas” Fees For Transactions So Low?* (2024). <https://solanacompass.com/statistics/fees>
- Suri, V. K., Elia, M., & van Hillegersberg, J. (2017). *Software Bots—The Next Frontier for Shared Services and Functional Excellence* (pp. 81–94).
https://doi.org/10.1007/978-3-319-70305-3_5
- Syed, R., Suriadi, S., Adams, M., Bandara, W., Leemans, S. J. J., Ouyang, C., ter Hofstede, A. H. M., van de Weerd, I., Wynn, M. T., & Reijers, H. A. (2020). Robotic Process Automation: Contemporary themes and challenges. *Computers in Industry*, 115, 103162.
<https://doi.org/10.1016/J.COMPIND.2019.103162>
- Syed, R., & Wynn, M. T. (2020). How to Trust a Bot: An RPA User Perspective. *Lecture Notes in Business Information Processing*, 393 LNBIP, 147–160.
https://doi.org/10.1007/978-3-030-58779-6_10/TABLES/1
- Tang, S., Zheng, J., Deng, Y., & Cao, Q. (2021). Resisting newborn attacks via shared Proof-of-Space. *Journal of Parallel and Distributed Computing*, 150, 85–95. <https://doi.org/10.1016/j.jpdc.2020.12.011>
- Textractor Documentation—Amazon-textract-textractor 1.0.0 documentation*. (2022). <https://aws-samples.github.io/amazon-textract-textractor/index.html>
- Thekkethil, M. S., Shukla, V. K., Beena, F., & Chopra, A. (2021). Robotic Process Automation in Banking and Finance Sector for Loan Processing and Fraud Detection. *2021 9th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*, 1–6. <https://doi.org/10.1109/ICRITO51393.2021.9596076>
- Thomas, C. G. (2021). *Research Methodology and Scientific Writing*. Springer International Publishing. <https://doi.org/10.1007/978-3-030-64865-7>

- Tian, Z., Li, M., Qiu, M., Sun, Y., & Su, S. (2019). Block-DEF: A secure digital evidence framework using blockchain. *Information Sciences*, 491, 151–165. <https://doi.org/10.1016/j.ins.2019.04.011>
- Total cost of ownership for blockchain solutions*. (2019).
- Touzou, M. E., & Garuz, J. T. (2020). *Bachelor's degree in Business Administration and Management*.
- Understanding Blockchain Latency and Throughput*. (2022, July 11). Paradigm. <https://www.paradigm.xyz/2022/07/consensus-throughput>
- Upadhyay, T., Chauhan, S., Sinha, U., Upadhyay SCSE, T., Chauhan SCSE, S., & Sinha SCSE, U. (2021). *Invoice Processing Automation*.
- Venable, J. R., Pries-Heje, J., & Baskerville, R. (2017). *Choosing a Design Science Research Methodology*.
- Weston, G. (2023, February 24). *What is Proof of History and how does it work?* 101 Blockchains. <https://101blockchains.com/proof-of-history/>
- Whiteboard, L. W. (2023, May 7). All Major Blockchain Consensus Algorithms Explained. *Medium*. https://medium.com/@learnwithwhiteboard_digest/all-major-blockchain-consensus-algorithms-explained-6934b4f5d47a
- Why Automation Matters: A Survey Study of the Modern Accounts Payable Department*. (n.d.). Retrieved April 29, 2024, from <https://www.ipayables.com/wp-content/uploads/2020/03/Why-Automation-Matters.pdf>
- Wieringa, R. J. (2014). Single-Case Mechanism Experiments. In R. J. Wieringa (Ed.), *Design Science Methodology for Information Systems and Software Engineering* (pp. 247–267). Springer. https://doi.org/10.1007/978-3-662-43839-8_18
- Willcocks, L. (2020). Robo-Apocalypse cancelled? Reframing the automation and future of work debate. *Journal of Information Technology*, 35(4), 286–302.

https://doi.org/10.1177/0268396220925830/ASSET/IMAGES/LARGE/10.1177_0268396220925830-FIG1.JPEG

- Willcocks, L., Lacity, M., & Craig, A. (2017). *Journal of Information Technology Teaching Cases*. 1–12. <https://doi.org/10.1057/s41266-016-0016-9>
- Wohlin, C. (2014). Guidelines for snowballing in systematic literature studies and a replication in software engineering. *ACM International Conference Proceeding Series*. <https://doi.org/10.1145/2601248.2601268>
- Wojciechowska-Filipek, S. (2019a). Automation of the process of handling enquiries concerning information constituting a bank secret. *Banks and Bank Systems*, 14(3), 175–186. [https://doi.org/10.21511/bbs.14\(3\).2019.15](https://doi.org/10.21511/bbs.14(3).2019.15)
- Wojciechowska-Filipek, S. (2019b). “Automation of the process of handling enquiries concerning information constituting a bank secret” NUMBER OF REFERENCES 28 NUMBER OF FIGURES 3 NUMBER OF TABLES 0 Automation of the process of handling enquiries concerning information constituting a bank secret. *Banks and Bank Systems*, 14. [https://doi.org/10.21511/bbs.14\(3\).2019.15](https://doi.org/10.21511/bbs.14(3).2019.15)
- Word Invoice Templates. (2024). https://invoicer.ai/word-invoice-templates?_qss=
- Xiao, B., Jin, C., Li, Z., Zhu, B., Li, X., & Wang, D. (2022). Proof of Importance: A Consensus Algorithm for Importance Based on Dynamic Authorization. *IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, 510–513. <https://doi.org/10.1145/3498851.3499007>
- Yakovenko, A. (2020, May 6). Proof of History: A Clock for Blockchain. *Solana*. <https://medium.com/solana-labs/proof-of-history-a-clock-for-blockchain-cf47a61a9274>
- Zemánková, A. (2019). *Artificial Intelligence and Blockchain in Audit and Accounting: Literature Review*.

ملخص الدراسة

مع تقدم الثورة الصناعية الرابعة والتوجه نحو بناء عمالة رقمية، أصبحت الأتمتة والأمن السيبراني من الأولويات الضرورية للمنشآت التجارية. تطبيقات أتمتة العمليات الروبوتية و تطبيقات الأتمتة الذكية المدعومة بتقنيات الذكاء الاصطناعي هي أهداف رخوة وسهلة للهجمات الإلكترونية حيث أن تلك التطبيقات تتعامل بشكل مباشر مع البيانات الشخصية الحساسة. تطبيقات الأتمتة الذكية تعاني من ضعف في عمليات التدوين والتدقيق، بحيث يستطيع أي مهاجم سيبراني أن يستغل هذه الثغرة لتزوير أو حذف ملفات تدوين العمليات بعد ارتكاب جريمة تسرب المعلومات أو الاحتيال الإلكتروني. تقدم هذه الأطروحة إطار عمل لتطبيق أتمتة ذكية مبني على تقنية البلوكتشين (BCIA) للحد من خطورة جرمي تسرب المعلومات والاحتيال الإلكتروني. يقوم تطبيق (BCIA) بأتمتة مهمات معالجة دفع الفواتير عبر تقنيات الذكاء الاصطناعي و بالإضافة إلى حماية سجلات العمليات وأرصدة الدفع من خلال حفظها على شبكة البلوكتشين. وأما عن منهاج البحث المعتمد في هذه الأطروحة، تم استخدام منهاج علوم التصميم (DSRM) لرسم مسار تطوير وتقييم إطار العمل (BCIA). وعلى مستوى التجارب التي أجريت في هذه الطروحة، فتم الاعتماد على وضع سيناريوهات تحاكي عمل مؤسسة تجارية في معالجة الفواتير باستخدام تطبيق (BCIA) للأتمتة الذكية مع ضمان حمايتها من تسرب المعلومات والاحتيال الإلكتروني. ونجح التطبيق في معالجة الفواتير وحفظ سجلات العمليات وأرصدة الدفع على شبكة البلوكتشين عند محاكاة عمل مؤسسة صغيرة (50 و 100 عملية) و متوسطة (500 عملية) وكبيرة (1000 عملية). وعند المقارنة بين إطار عمل (BCIA) الذي يعتمد على شبكات البلوكتشين العامة مع إطار (Block-DEF) و (BCFL) اللذان يعتمدان على شبكات البلوكتشين الخاصة في إنشاء الأدلة الإلكترونية بحفظها على شبكة البلوكتشين، تبين بأن إطار عمل وتطبيق (BCIA) حقق نجاحاً في حفظ سجلات العمليات وأرصدة الدفع من الحذف أو التزوير مما جعل (BCIA) مثلاً رائداً لاستخدام شبكات البلوكتشين العامة في هذا المجال. وفي تجربة أخرى تمت مقارنة أداء ثلاث شبكات بلوكتشين عامة وهي (Sepolia ETH, Solana Devnet, BNB Chain) من حيث إنتاجية الشبكة وحجم البلوك و تكلفة العمليات ووقت الاستجابة لكل عملية والمدة المطلوبة لإنهاء العمليات. وأظهرت نتائج التجربة بأن شبكة (BNB Chain) كانت الأفضل أداءً والخيار الأمثل للمؤسسات التجارية بمختلف أحجامها حيث حققت متوسط إنتاجية الشبكة بـ 47 عملية في الدقيقة الواحدة ومتوسط مدة إنهاء العمليات بـ 9.3 ثواني فقط وتكاليف مالية منخفضة بحيث تكلفة معالجة 1000 فاتورة لم تتجاوز 4.20 دولار أمريكي.

جامعة البحرين
كلية تقنية المعلومات



تطبيق للأتمتة الذكية لمحاربة الاحتيال الإلكتروني وتسرب البيانات في عمليات
التجارة البينية بين المنشآت مبني على تقنية البلوك تشين

أطروحة مقدمة كجزء من متطلبات الحصول على درجة الدكتوراة في علوم الحوسبة والمعلومات

إعداد:

يعقوب سلمان محمد السليس

20190179

بإشراف

الدكتور مازن علي

(أستاذ مشارك)

مملكة البحرين

يونيو 2024