PGCOMP - Programa de Pós-Graduação em Ciência da Computação
Universidade Federal da Bahia (UFBA)
Av. Milton Santos, s/n - Ondina
Salvador, BA, Brasil, 40170-110

https://pgcomp.ufba.br
pgcomp@ufba.br

Requesting food online is daily present in the lives of people, however, it is necessary to send delivery information to different restaurant systems and sometimes these restaurants share customer-sensitive information with their partners without the authorization of the customer creating a security risk to customers' data. Also, in most cases when a person travels to a different country and wants to request food online, it is necessary to use the local currency, requiring a currency exchange operation. Blockchains revolutionized the online payment method and intrinsically safely eliminated central authoritative entities with Bitcoin as the first blockchain utilized on a large scale. However, the scientific literature recognizes the difficulties in such technology as scalability flaws in the number of transactions executed per second on Bitcoin, high transaction time, and also the high fee per transaction, beyond the limitation of Script language. The Ethereum Blockchain was raised as a Bitcoin alternative that allowed the development of more complex applications, and despite the increase in the default number of transactions per second, it is still away from payment method solutions such as credit cards, furthermore, it had for a long period a high fee per transaction. In this thesis, we studied and proposed blockchain as a payment method for retail transactions and we created a Proof-Of-Concept (PoC) for restaurants utilizing the Lisk blockchain Software Development Kit (SDK). The Lisk Restaurant PoC represents an integration of the Lisk blockchain in the food industry allowing customers to order food fast and safely through a sidechain, an exclusive blockchain with specific custom transaction types for restaurants, eliminating third parties, reducing fees, and utilizing a single crypt-asset around the world, LSK. We performed an empirical study on our proposed solution to evaluate the performance of the number of transactions per block, transactions fee, audit ability, scalability of solution, the sending of a data message in transactions safely, and guaranteeing privacy. The results were utilized to compare with known information in literature from blockchains Bitcoin, Ethereum, and Multichain. The specific blockchain transaction types utilized in Lisk Restaurant allow guarantee data privacy for its customers allowing only transaction recipients to access sender data. Hence, through research, revision of previous research findings, and experiments of performance evaluation it was observed that the restaurant sidechain solution demonstrated the efficiency of Blockchain technology in providing global services less costly than regular ways of payment, or even other blockchains as Bitcoin or Ethereum through lower transactions fee, and respecting the privacy of customers. Furthermore, the new solution demonstrated an increase in the maximum number of Food transactions capacity in a single block, and better transaction scalability through sidechains, these accept transactions related to their business differently from blockchains such as Bitcoin or Ethereum that don't have sidechains as they execute transactions with distinct business needs, superior to the first version of PoC Lisk Restaurant, surpassing by twice in the number of transactions capacity in a single block. Finally, experiments were performed with the Lightning network, an off-chain technology that runs on top of Bitcoin offering a scalability solution for Bitcoin. The evaluation of the results of the experiments of the Lisk sidechain solution and Lightning network show the differences between each technology and highlight the advantages and disadvantages of each one. Hence, it was observed that the Lightning network can be utilized in the retail industry but without the possibility of customized transaction costs.

MSC | 162 | 2023

A sidechain platform for restaurant e-commerce transactions

Davi Lima Alves

UFBA

A sidechain platform for restaurant e-commerce transactions

Davi Lima Alves

Dissertação de Mestrado

Universidade Federal da Bahia

Programa de Pós-Graduação em
Ciência da Computação

Agosto | 2023

Universidade Federal da Bahia
Instituto de Computação

Programa de Pós-Graduação em Ciência da Computação

A SIDECHAIN PLATFORM FOR RESTAURANT E-COMMERCE TRANSACTIONS

Davi Lima Alves

DISSERTAÇÃO DE MESTRADO

Salvador
22 de Agosto de 2023

DAVI LIMA ALVES

A SIDECHAIN PLATFORM FOR RESTAURANT E-COMMERCE TRANSACTIONS

Esta Dissertação de Mestrado foi apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal da Bahia, como requisito parcial para obtenção do grau de Mestre em Ciência da Computação.

Orientador: Manoel Gomes de Mendonça Neto

Salvador
22 de Agosto de 2023

Sistema de Bibliotecas - UFBA

Lima Alves

Lima Alves, Davi.

A sidechain platform for restaurant e-commerce transactions / Davi Lima Alves – Salvador, 2023.

177p.: il.

Orientador: Prof. Dr. Manoel Gomes de Mendonça Neto.

Dissertação (Mestrado) – Universidade Federal da Bahia, Instituto de Computação, 2023.

1. Lisk. 2. Blockchain. 3. DPoS. I. Gomes de Mendonça Neto, Manoel. II. Universidade Federal da Bahia. Instituto de Computação. III Título.

CDU – 004

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DA BAHIA
INSTITUTO DE COMPUTAÇÃO
PGCOMP - Programa de Pós-Graduação em Ciência da Computação
http://pgcomp.ufba.br

"A SIDECHAIN PLATFORM FOR RESTAURANT E-COMMERCE TRANSACTIONS"

Davi Lima Alves

Dissertação apresentada ao Colegiado do Programa de Pós-Graduação em Ciência da Computação na Universidade Federal da Bahia, como requisito parcial para obtenção do Título de Mestre em Ciência da Computação.

Banca Examinadora

gov.br Documento assinado digitalmente
MANOEL GOMES DE MENDONCA NETO
Data: 03/09/2023 22:51:14-0300
Verifique em https://validar.iti.gov.br

Prof. Dr. Manoel Gomes de Mendonça Neto
(Orientador PGCOMP)

gov.br Documento assinado digitalmente
ALLAN EDGARD SILVA FREITAS
Data: 04/09/2023 11:25:50-0300
Verifique em https://validar.iti.gov.br

Prof. Dr. Allan Edgard Silva Freitas (IFBA)

gov.br Documento assinado digitalmente
ROBESPIERRE DANTAS DA ROCHA PITA
Data: 04/09/2023 19:06:25-0300
Verifique em https://validar.iti.gov.br

Prof. Dr. Robespierre Dantas da Rocha Pita (PGCOMP)

gov.br Documento assinado digitalmente
LEOBINO NASCIMENTO SAMPAIO
Data: 04/09/2023 16:25:52-0300
Verifique em https://validar.iti.gov.br

Prof. Dr. Leobino Nascimento Sampaio (PGCOMP)

Prof. Dr. Cássio Vinicius Serafim Prazeres (PGCOMP)

DEDICO ESSE TRABALHO MINHA FAMLIA QUE ME APOIOU EM MINHA VIDA.

ACKNOWLEDGEMENTS

Agradeço a todos que me apoiaram desde o início do mestrado em Ciência da Computação na Universidade Federal da Bahia. Agradeço aos professores que se dedicaram em ensinar conteúdo de qualidade durante o mestrado. Agradeço à minha família, minha mãe e meu irmão por toda atenção, pelo suporte, em todos os momentos da minha vida e durante o mestrado, ao meu pai por ter lido versão em português de texto antes mesmo de qualificação e principalmente ao meu filho maravilhoso que tenho tanto orgulho. Quanto ao mestrado agradeço à Universidade Federal da Bahia, e também à comunidade da blockchain Lisk, aos membros da Lisk, aos CEOs dessa blockchain que acompanhei desde a versão 1, aos pesquisadores que leram meus trabalhos ciéntificos e à IOP, ACM, IntechOpen, por terem publicado meus trabalhos científicos, mas além disso terem me convidado para webinars de entendimento procedimento peer review, ou ainda por ter recebido 1 ano de revista científica ACM Qualis A1 gratuitamente, ou também pelo incentivo de ser convidado à diversos conferências e principalmente de continuar escrevendo. Obrigado!

Dedico este trabalho aos meus pais por me suportarem desde o inicio. Ao meu filho Henrique, que me incentivou e foi sempre compreensivo. Ao meu irmao Danilo por suportar e acreditar no objetivo da conclusao deste trabalho.

—DAVI LIMA ALVES (NOTA)

RESUMO

Pedir comida *online* está presente diariamente na vida das pessoas, entretanto, este procedimento exige o envio de dados de clientes para diferentes sistemas de restaurantes, e muitas vezes esses restaurantes compartilham informações sensíveis com seus parceiros sem autorização, criando um risco à segurança ao compartilhar tais dados. Ainda, em muitos casos quando uma pessoa viaja para um país diferente de sua origem e deseja pedir comida é exigida a transação de câmbio para moeda local que tem custo elevado. As *blockchains* revolucionaram o modo de fazer pagamentos online e intrinsecamente de forma segura eliminando a entidade centralizadora e autorizativa, a exemplo da Bitcoin, primeira *blockchain* utilizada em larga escala. Entretanto, a literatura científica registra a dificuldade diante da alta frequência de transações na Bitcoin, alto tempo de transações, e, também, seu alto custo por transação além da limitação da linguagem Script para desenvolvimento de aplicações. Nesse passo a *blockchain* Ethereum surge como alternativa a Bitcoin, num ambiente de aplicações mais complexas com maior frequência de aplicações e número de transações por segundo, mas, ainda distante da performance de soluções como cartão de crédito, o que desfavorece esta "cadeia de blocos" ante soluções tradicionais de pagamento como o cartão de crédito, além disso teve por um grande período altas taxas de custo por transação. Do estudo deste contexto nós propomos o modelo de pagamento de transações para o varejo por meio da *blockchain* a partir de Prova de Conceito (POC) voltada para restaurantes utilizando o SDK da *blockchain* Lisk. A POC Lisk Restaurante representa uma integração da Lisk *blockchain* na indústria de alimentos permitindo clientes comprarem comida de forma rápida e segura através de *sidechains* - *blockchains* exclusivas para transações especiais tipo-restaurantes, sem intermediação e com menores taxas ao utilizar um único cripto ativo no mundo: o LSK. Realizamos estudo empírico da solução proposta no qual é avaliada o desempenho de transações em bloco, taxas de transação, auditabilidade, escala de solução, privacidade e segurança no envio de mensagens em transações. Os resultados foram utilizados na comparação com dados da literatura das *blockchains* Bitcoin, Ethereum, e Multichain. Os resultados apontam que, as transações especiais utilizadas na Lisk Restaurante permitem garantir privacidade de dados para seus clientes, pois, apenas ao recipiente de uma transação é possível acessar dados sensíveis de remetente. Assim, através de pesquisa, revisão da literatura e experimentos de avaliação de desempenho foi observado que a solução de *sidechain* para restaurantes demonstrou eficiência da tecnologia *blockchain* em prover serviços globais de forma menos custosa do que meios de pagamentos comuns; também, quando comparadas suas taxas de transações com outras *blockchains* como Bitcoin ou Ethereum, pois, seus valores são menores e mantida a privacidade de seus clientes. Ainda, a nossa solução demonstrou aumento na capacidade de transações *Food* em um único bloco e melhor escalabilidade por meio de *sidechains*, as quais, aceitam transações específicas ao negócio envolvido o que difere de blockchains únicas como Bitcoin ou Ethereum que executam transações de diversas finalidades, superando também à primeira versão da Prova de Conceito (PoC) Lisk Restaurante, ultrapassando em duas vezes a capacidade de número de transações ou frequência em um bloco. Finalmente, ainda foi realizado experimentos com a Lightning network, tecnologia off-chain que funciona sobre a Bitcoin sendo uma solução de escalabilidade para a Bitcoin. A avaliação de resultados de experimentos da solução sidechain Lisk e da Lightning network demonstraram as diferenças de cada tecnologia e foi enaltecida as vantagens e desvantagens de cada uma. Assim, foi observado que a tecnologia Lightning network também pode ser utilizada na indústria de varejo, mas sem a possibilidade de personalizar custos de transação.

Palavras-chave: LISK, BLOCKCHAIN, BITCOIN, SIDECHAIN, PROVA DE CONCEITO, DESEMPENHO, PRIVACIDADE, CUSTO, LIGHTNING NETWORK

ABSTRACT

Requesting food online is daily present in the lives of people, however, it is necessary to send delivery information to different restaurant systems and sometimes these restaurants share customer-sensitive information with their partners without the authorization of the customer creating a security risk to customers' data. Also, in most cases when a person travels to a different country and wants to request food online, it is necessary to use the local currency, requiring a currency exchange operation. Blockchains revolutionized the online payment method and intrinsically safely eliminated central authoritative entities with Bitcoin as the first blockchain utilized on a large scale. However, the scientific literature recognizes the difficulties in such technology as scalability flaws in the number of transactions executed per second on Bitcoin, high transaction time, and also the high fee per transaction, beyond the limitation of Script language. The Ethereum Blockchain was raised as a Bitcoin alternative that allowed the development of more complex applications, and despite the increase in the default number of transactions per second, it is still away from payment method solutions such as credit cards, furthermore, it had for a long period a high fee per transaction. In this thesis, we studied and proposed blockchain as a payment method for retail transactions and we created a Proof-Of-Concept (PoC) for restaurants utilizing the Lisk blockchain SDK. The Lisk Restaurant PoC represents an integration of the Lisk blockchain in the food industry allowing customers to order food fast and safely through a sidechain, an exclusive blockchain with specific custom transaction types for restaurants, eliminating third parties, reducing fees, and utilizing a single crypt-asset around the world, LSK. We performed an empirical study on our proposed solution to evaluate the performance of the number of transactions per block, transactions fee, audit ability, scalability of solution, the sending of a data message in transactions safely, and guaranteeing privacy. The results were utilized to compare with known information in literature from blockchains Bitcoin, Ethereum, and Multichain. The specific blockchain transaction types utilized in Lisk Restaurant allow guarantee data privacy for its customers allowing only transaction recipients to access sender data. Hence, through research, revision of previous research findings, and experiments of performance evaluation it was observed that the restaurant sidechain solution demonstrated the efficiency of Blockchain technology in providing global services less costly than regular ways of payment, or even other blockchains as Bitcoin or Ethereum through lower transactions fee, and respecting the privacy of customers. Furthermore, the new solution demonstrated an increase in the maximum number of Food transactions capacity in a single block, and better transaction scalability through sidechains, these accept transactions related to their business differently from blockchains such as Bitcoin or Ethereum that don't have sidechains as they execute transactions with distinct business needs, superior to the first version of PoC Lisk Restaurant, surpassing by twice in the number of transactions capacity in a single block. Finally, experiments were performed with the Lightning network, an off-chain technology that runs on top of Bitcoin offering a scalability solution for Bitcoin. The evaluation of the results of the experiments of the Lisk sidechain solution and Lightning network show the differences between each technology and highlight the advantages and disadvantages of each one. Hence, it was observed that the Lightning network can be utilized in the retail industry but without the possibility of customized transaction costs.

Keywords: LISK, BLOCKCHAIN, BITCOIN, SIDECHAIN, PROOF-OF-CONCEPT, PERFORMANCE, PRIVACY, TRANSACTION FEE, LIGHTNING NETWORK

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

API	Application Programming Interface	60
BFT	Byzantine Fault Tolerance	31
CAP	Consistency, Availability, Partition Tolerance	78
CCC	Cross Chain Certificates	37
CCM	Cross Chain Messages	37
CCU	Cross Chain Updates	37
CES	Content Extraction Signature	44
DDoS	Distributed Denial-of-Service	21
DoS	Denial-of-Service	21
DPoS	Delegated Proof of Stake	99
ECDSA	Elliptic Curve Digital Sign Algorithm	12
ERC20	Ethereum Request For Comments 20	42
HTLC	Hashed timelock contract	30
HTTP	Hyper Text Transport Protocol	65
IPC	Interpersonal communication	56
JSON	JavaScript Object Notation	66
PG	Payment Gateway	41
P2P	peer-to-peer	45
PBFT	Practical Byzantine Fault Tolerance	20
PoC	Proof-Of-Concept	98
PoS	Proof-of-Stake	20
PoW	Proof-of-Work	41
SDK	Software Development Kit	99
SPV	Simplified Protocol Verification clients	26
SSL	Secure Socket Layer	60
UTXO	Unspent Transaction Output	90
VM	Virtual machine	81
WS	Web Socket	60

Chapter 1

INTRODUCTION

Blockchain is a technology capable of grouping transactions into blocks, where each block created is identified by a unique cryptographic hash, and each block references the cryptographic hash of the previous block forming a chain of blocks as the name suggests. The chain of blocks initiates with a genesis block and only appends information, furthermore because its composition is immutable and secure. The blockchain runs on any node Virtual machine (VM)s that want to participate in this peer-to-peer (P2P) network. The blockchain permits any participant to perform transactions without a trusted authority. To perform a transaction each participant utilizes a digital address (usually a hash from a cryptographic public key) (ZOHAR, 2015). In a blockchain, users are not identified by names but by digital addresses. This is important to protect users' privacy.

Blockchain technology has been successfully applied to several problem areas requiring security and privacy without the use of a central trust authority (DAMSGAARD, 2022). For example, Audius is a decentralized application and streaming platform that allows its users to share music, it was created to give back power to content creators through the use of Blockchain technology. Solana blockchain is behind Audius, and it is shown to be particularly good at acting as a platform for music sharing where artists can upload their music and users can listen to them without any infringement. The use of blockchain eliminates the middleman, reducing fees and empowering its participants, a common requirement for modern digital platforms (DAMSGAARD, 2022). Another example of good utilization of blockchain technology is the Brazilian Startup called Arabyka which specializes in traceability utilizing blockchain technology and utilizes the technology on a supply chain to monitor many phases of the coffee including season, harvest types, drying, day of exportation, and much other information to export coffee to Japan. It was mentioned by the authors of Arabyka (AGRO, 2022) that the immutability, transparency, and decentralization characteristics offered by blockchain are fundamental characteristics to guarantee food safety. The most famous blockchain in the world is Bitcoin (NAKAMOTO, 2009). In fact, Bitcoin since 2009 allowed several pseudo-users to transfer the BTC cryptocurrency between the participants in its network without a trusted authority.

(LI et al., 2022) states that blockchain innovated in how digital information is stored, verified, and exchanged, and was inherently designed and developed to create secure, reliable, and transparent business processes for enterprises. Also, they present a survey that reveals that the global blockchain market size is expected to grow from USD 4.9 billion in 2021 to USD 67.4 billion in 2026. Furthermore, organizations have started to explore and experiment with blockchain's potential by developing blockchain applications, therefore, the proper choice of a "good" blockchain platform becomes vital.

It is also essential to consider how long it takes for new information to be appended to a blockchain. New information is only available when a block is included in a blockchain. Otherwise, the state of all nodes that compose the blockchain would be the same. Each blockchain can provide a different approach for having a new block into the chain, directly impacting the duration of the transactions executed over the network. For example, the blockchain platform used in Bitcoin requires 10 minutes for a block to be included in a chain. On the other hand, the Lisk (LISK, 2021b) blockchain platform utilizes a Delegated Proof of Stake (DPoS) with Lisk-BFT consensus, which is much faster than Bitcoin's approach (XIAO et al., 2020). The approach used in Lisk only requires 10 seconds for a block to be included in a blockchain.

This dissertation aims to experiment, observe, and analyze the performance of the use of blockchain in the retail industry as it represents a growing online industry, also it will verify the cost, privacy, and scalability of the solutions evaluated.

1.1 THESIS PROBLEM STATEMENT

Dissertation statement (Problem). *As written in the scientific literature related to traditional blockchains, currently, exists a necessity for higher throughput to execute more transactions in less time, a necessity of reducing fee costs of a transaction, and a necessity of preserving the privacy of sensitive user data that execute transactions on-chain in a blockchain network. Performance is a challenge in public blockchains because of the necessity of achieving agreement in distributed consensus mechanisms to include a new block by network nodes in the blockchain. Blockchain protocol parameters such as block size, and block interval (time to create a block) are relevant for determining the processing time of a transaction. Also, the storage capacity of a node is determined by the block parameters that cause higher or lower throughput. In the blockchain is possible to audit the ledger of any member of the network in such a way that preserving user-sensitive data is crucial for participants that want to utilize the blockchain network and keep their sensitive data private. Beyond that, the fee costs to execute a transaction can be determinant for utilizing a blockchain network in a specific scenario.*

For this problem, we will verify the feasibility of blockchain as a payment method platform for the retail industry providing fast and safe solutions allowing customers to request and buy food online with reduced fees, eliminating intermediaries, preserving customer-sensitive data on-chain, and utilizing the LSK crypto-asset.

1.2 THESIS OBJECTIVES

1.2.1 Main objective

This work aims to develop and evaluate the performance of a blockchain on-chain solution (transaction data are stored in the blockchain), in fact, a sidechain, an exclusive blockchain that can customize configurations and transactions. Therefore, the development and evaluation will be performed on the Proof-Of-Concept (PoC) sidechain solution based on blockchain as a computational system and payment method for the retail industry based in Lisk blockchain SDK. The solution allows customers to request food in place or online from restaurants that participate in the restaurant sidechain solution preserving their sensitive data utilizing a cryptocurrency recognized worldwide and reducing fees by removing intermediaries from a Payment Gateway (PG). Also, it is important to state what is **not considered as a scope of this work**:

- performing a market analysis
- verifying the most commonly adopted solution by the market
- verifying the quality of the implementation

1.2.2 Main specific objectives

- Creation of low-cost sidechain Proof-Of-Concept (PoC) for e-commerce
- Creation of custom transaction types for example Food transaction type to preserve customer-sensitive data sent to the recipient of a transaction and store the data encrypted in the blockchain
- Creation of Profile transaction type for customer preserving the latest customer sensitive data
- Creation of Menu transaction type to list products from retail
- Creation of libraries to allow integration to the retail web application with the proposed sidechain
- Evaluate the performance of the proposed sidechain solution measuring throughput as the total capacity of the transactions in a block per second
- Verify scalability and cost of the solution

1.3 RESEARCH QUESTIONS

This thesis aims to answer the following questions regarding the solution efficiency proposed:

- Is it possible to create a sidechain blockchain solution for the retail industry?

  Specific question:

- Does it guarantee data privacy?
- Does it have a low cost?
- Does it allow for public audits of transactions?
- Does it scale?
- What is its performance?
    - Does the increase in block size on the sidechain allow for a reduction in the time for forging a block on the restaurant sidechain?
    - Does the increase of block size on the sidechain or reducing the time for forging a block provide better throughput than the first version of the Restaurant sidechain (ALVES, 2021b)?

1.4 HYPOTHESIS

Dissertation statement (Hypothesis)**.** *We hypothesize that developing a sidechain could bring the necessary customization configuration for executing transactions in a reasonable time, ensuring the protection of user data privacy and reducing transaction fee costs in the retail industry.*

1.5 METHODOLOGY

This Section establishes a set of steps that compose the methodology defined in this work describing the goals and results of each step, Table 1.1. Each step is composed of a Goal, Task(s) to accomplish the goal, the Results generated, and when exists Publication(s) by the author of this work.

The evaluation of performance will be shown by experiments performed with the sidechain, the customer-sensitive data privacy can be verified by analysis of the Proof-Of-Concept (PoC) proposed and the custom transaction types, the reduced costs fees can be verified by transaction analysis, and scalability of solution can be verified by the design of solution defined in Chapter 4.

The bibliography revision of scientific literature was performed in diverse fields of computing and the main works were included in a matrix of related works [1]. Chapter 3 contains the matrix of related works with relevant papers, the search criteria with the most relevant search keywords, and the relevance of the most important papers. For performing such research it was utilized search tools such as Google Scholar, ACM, IEEE, IOP Publishing, Portal Periodicos, journals, and conferences recognized and not recognized by Qualis regarding the subject blockchain.

Following is a briefing description of the activities that compose the methodology:

1. **Perform bibliographic revision of literature:** Perform bibliography revision in the scientific literature regarding blockchain, and blockchain-based platforms

[1] Table A.1

Table 1.1 Methodology steps of this work

Goal	Task	Results	Publications
Understanding blockchain concepts and blockchain-based platforms for payment	Performing bibliographic revision of literature	Matrix of related works with more than 40 relevant papers. Narrow of research, theoretical and practical knowledge, and related works paper	(ALVES, 2020; ALVES, 2021b; ALVES, 2021a; ALVES; GREVE, 2021)
Investigate blockchain-based platforms for payment in different consensus mechanisms and perform a selection of technologies	Perform a study and elaborate a table of contents regarding the comparison between blockchain solutions and their boundaries	Definition of chosen solution	
Design of solution for secure low-cost payment blockchain-based platforms	Implementation of new custom transactions types and sidechain solution	Blockchain payment-based solution as sidechain	(ALVES, 2021b)
Identification of technologies and tools for performance evaluation in blockchains Performance evaluation of the solution	Verify tools of performance evaluation in diverse consensus mechanisms Verification of the scalability, performance, and reliability of the solution	Most tools allow performance evaluation in public blockchains with Proof-of-Work (PoW) or private blockchains but they lack accuracy or compatibility, then utilizing the real network for evaluation is better Tables of results with empirical analysis and conclusion	(ALVES, 2021b)

for low cost on public, hybrid, and private blockchains utilizing search tools and researching in scientific and non-scientific literature, sites. This task can be found in Chapters 1 and 2.

2. **Perform a study and elaborate a table of contents regarding the comparison between blockchain solutions and their boundaries:** Select technologies to support secure low-cost payment blockchain-based platforms and elaborate table of contents between them containing their cons and pros. Such a task can be found in Chapter 3.

3. **Implementation of new custom transactions types and sidechain solution:** Design the solution and implement the source-code logic containing the requirements of the solutions by creating and registering the new transaction types in the sidechain proposed.

4. **Verification of the scalability, performance, and reliability of solution:** In this task, a study was performed to understand and discover the available tools for performing performance evaluation or techniques for performing performance evaluation, therefore this step verify tools of performance evaluation in diverse consensus mechanisms. The survey from (FAN et al., 2020) was very important in determining the performance evaluation technique for this work. Hence, it was identified that an empirical analysis of the proposed sidechain solution and its network would be the best option for performing a performance evaluation. At the moment of the writing of this work lacks tools for performing performance evaluation for Delegated Proof of Stake (DPoS) consensus mechanism. Experiments were created with different scenarios utilizing the proposed solution for performing transactions. The experiments were based on the work from (AKBARI et al., 2020), they were observed, and evaluated, and a detailed analysis was described in Chapter 5.

The work from (AKBARI et al., 2020) included in the matrix of related works explained precisely in Chapter 3, Section 3.4, deserves highlight as it is very relevant to this dissertation, which proposes the modification of blockchain parameters such as block size and block creation time in the Bitcoin blockchain by experimenting with a simulator. Such characteristic is similar to the possible configurations in a Lisk sidechain that has a *DPoS* consensus mechanism. For evaluating the performance of the proposed solution some works are more relevant than others and deserve highlight as the works from (FAN et al., 2020) that is a blockchain evaluation survey demonstrating techniques for performance evaluation and possible tools to perform an evaluation when it is not possible to utilize test network or real network for experiments. Also, the simulators BlockSIM described in (ALHARBY; MOORSEL, 2020) was created for evaluating blockchain-like with PoW consensus mechanism allowing extension of the tool and implementation for different consensus mechanisms, or even the tool Blockbench described in (DINH et al., 2017) that is specialized for evaluating private blockchains. Finally, for comparison purposes, the scalability solution from Bitcoin called Lightning network was utilized in experiments, detailed in Chapter 5 to highlight the differences between technologies from the proposed PoC solution.

1.6 MAIN ACCOMPLISHMENTS

It was decided to create a Proof-Of-Concept (PoC) sidechain solution with custom transactions for the retail industry, and restaurants, performing an empirical study for evaluating the proposed solution of this work and measuring its performance with a validation technique that will be detailed described in Chapter 5. The experiments considered for evaluating the PoC solution are based on three main aspects or metrics throughput, waiting time, and stale blocks. Therefore, throughput measures the number of transactions inside a block per time, waiting time represents the time how long a user waits for its transaction to be included in a block, and stale blocks are related to keeping consistent information in the system. For the experiments, some parameters were configured and changed in the sidechain solution on each experiment for evaluation purposes. Changing the characteristics of each experiment that have a direct impact on the metrics evaluation of the solution is a technique described in the work from (AKBARI et al., 2020). The parameters changed are the number of users that performed a food request, block interval which is the time to create a block in seconds, and block payload size which is the total size capacity in KB of transactions included in a block. Furthermore, it was created 10 scenarios for evaluating the performance of the solution, and 40 experiments with more than 5000 transactions, also it was considered the geographic location of blockchain nodes in each experiment. Alternatively, we experimented with the Lightning Bitcoin network for comparison purposes with the proposed PoC solution. For that, we created 3 types of experiments that included payment of 40 Lightning invoices, 2 concurrent payments of Lightning invoices, and a withdrawal from Lightning to the Bitcoin network. A matrix of related works with more than 40 relevant works is also a contribution to this work. Therefore, the contribution of this work can be defined as follows:

Dissertation statement (Contribution)**.** *The solution created and the performance evaluation performed in this work show that blockchain technology offers an acceptable performance level for executing transactions in the proposed case study. The custom transactions through sidechains allowed necessary customization for reducing transaction costs, protecting customer-sensitive data in an on-chain solution, augmenting scalability, and enabling public audibility. The performance evaluation showed different perspectives regarding performance and security when blockchain parameters were configured in the experiments in the Lisk sidechain and highlighted configurations that can improve the throughput of the solution. The matrix of related works with more than 40 papers also is a contribution to this work.*

1.7 ORGANIZATION OF THIS DOCUMENT

The organization of this work is: Chapter 2 provides blockchain technology fundamentals required to understand Lisk Restaurant Sidechain, Bitcoin, and Lightning network. Chapter 3 describes related works, and highlights the matrix of related works, Chapter

4 explains the prototype design of the proposed Lisk Restaurant solution. Chapter 5 describes the evaluation scenarios explored to analyze the solutions proposed, Chapter 6 covers the validation of privacy, costs, and scalability by the design of the proposed solution and Chapter 7 is the conclusion of this work and explores possible topics for future work.

Chapter 2

BLOCKCHAIN FUNDAMENTALS

This chapter introduces important concepts utilized as part of the blockchain technology in Section 2.1, then blockchain fundamental concepts are provided in Sections 2.2 and 2.3, setting up the basis for explaining two well-known cryptocurrencies, BTC and LSK, which are based on the Bitcoin and Lisk blockchains, respectively, presented in Sections 2.4 and 2.6. Although there are several other types of blockchains, the focus on these two technologies is twofold. First, Bitcoin is the most famous use case, responsible for introducing several of the concepts used in other blockchain technologies. Second, Lisk is the blockchain choice for supporting the present work, as it favors the development of decentralized applications. The reasons for this choice will be clearer in the next chapter. Section 2.5 introduces the Lightning network, an off-blockchain solution executed above the Bitcoin network for processing micropayments in channels. Once, the micropayments are processed and agreed upon by the participants in a channel then the micropayments are sent to the Bitcoin network to be included in a block. The chapter ends with a chosen platform, differences of platforms, and interesting properties of blockchain in Sections 2.7, 2.8, 2.9.

2.1 CONCEPTS UTILIZED AS A BASE FOR BLOCKCHAIN TECHNOLOGY

The peer-to-peer (P2P) network is present in blockchain and many other distributed computer systems. Before its utilization in blockchain, P2P utilization can be found in many distributed system programs but it was the famous Napster program that highlighted its benefits of scalability, fault tolerance, and decentralization. P2P are distributed without any central control and their composition is based on overlay networks, in which nodes are connected by logical links, which sometimes can utilize many physical links to achieve a connection between two nodes (LUA et al., 2006).

P2P allows different peers to store and share the same data content in a network, hence, allowing other peers to access such data content connecting not to a single peer but any peer that has such data content, hence, it includes decentralization. Also, it includes fault tolerance, for example, if a peer that holds data content disconnects from the network then another peer that has such data content and is connected to the network can share the data requested with the requester. Furthermore, it can increase scalability as many peers request the data content then P2P can distribute the requests between the peers that have such data content, and as many peers hold the same data content then more requests to such data content the system will be able to respond faster as it affects directly the performance to retrieve information in the system (LUA et al., 2006).

There are two types of overlay networks, structured and unstructured. The structured overlay network is based on a structured graph and each data item present is associated with a peer based on a key. The unstructured overlay network organizes peers in a random graph in a simple way or hierarchically, also it supports complex queries for content, however, the search for content is performed locally on each peer and this characteristic makes the search less efficient than structured overlay networks (LUA et al., 2006).

Each data item key stored by a peer in P2P is based on a hash that makes the item unique on each peer, such hash was acquired after applying a hash function to the entire data content. Such a cryptographic hash function is a function that can compress data of arbitrary length to bit-strings of a fixed length, for which it is computationally infeasible to find two different data utilized in compression that are mapped by the function to the same hash value, however, if it happens it is called a collision(ZOHAR, 2015). Therefore, if a bit is changed in the data content then applying a cryptographic hash function on it will generate an entirely different hash as a result. This characteristic helps in the safe generation of unique data item keys, in many situations based on public and private keys, and then in providing the authenticity of the data content downloaded by a peer that can apply an integrity verification in the data content downloaded (LUA et al., 2006).

Despite its decentralized architecture P2P can suffer from security problems, for example, man-in-middle attacks in which malicious peers could return wrong data objects from lookup queries or even respond with false information eavesdropping on the communication between other nodes. Other types of attacks are also susceptible in P2P as Sybil attacks, which difficult to certify the identity of a peer in the network in the presence of several malicious peers for example (LUA et al., 2006). Hence, identifying uniquely and safely a peer in the network is very important.

Nevertheless, a reliable computer system must be able to cope with failures of its components in such a way it determines whether is possible to proceed or stop entirely. The paper from (LAMPORT; SHOSTAK; PEASE, 2002) proposed a solution expressed abstractly with the utilization of a group of generals of the Byzantine army around an enemy city they need to decide if they will invade or not the city. However, the generals are geographically split and they can communicate between themselves only by sending messengers, and only with a common plan they can win. In this scenario, it is possible to achieve agreement only if more than two-thirds of the generals are loyal. It means there is the possibility that one or some of the generals are traitors, behaving maliciously, and such a traitor can confound other general(s). As much increases the number of generals in the scenario more communication will be necessary to achieve an agreement and a communication overhead could reduce performance for it. Beyond that, the necessity of uniquely identifying each general message is imperative as also establishing communication between them to achieve consensus.

All these concepts of P2P, cryptographic hash function, consensus and the security they can provide made them suitable for their utilization in blockchain technology, which can have a great frequency of nodes that connect and disconnect the network and an agreement regarding the data content stored should be achieved by its peers that can behave honestly (desired) or behave maliciously (undesired). Furthermore, based on the technologies it compounds blockchain users are identified as pseudo-users, which guarantees user privacy, and messages exchanged by them are signed by the utilization of a unique combination of hash, public keys, and private keys.

2.2 BLOCKCHAIN CONCEPTS AND THE DISTRIBUTED LEDGER DATA STRUCTURE

The blockchain has a distributed data structure that behaves like a ledger registering all transactions. The information is synchronized at all nodes, which keep the same version of the blockchain via a consensus protocol. In Bitcoin and blockchain in general the nodes that run such a protocol verify and store each received transaction information so that the same data is locally replicated. Transactions are grouped into blocks and each newly created block contains the cryptographic hash of the previous one, which is unique. As the name suggests, this structure consists of a chain of blocks, forming an incremental log that contains all transactions that have ever occurred.

The responsible for block creations are called miners in Bitcoin and forgers in Lisk. These block creators, and only them, can include a block in the network. The way a block is validated in each blockchain network may vary. In general, it can be said that after a block creation, the miner/forger sends it for validation by other nodes in the network. If the block is validated by a majority of the nodes in the network, the block is accepted in the blockchain.

The first ever block created is called the Genesis block. All transactions in the blockchain are stored in the ledger and blocks, starting from the Genesis block. This blockchain structure allows for auditing all transactions as a block point to the previous one. Further, if one reads the transactions from the first block until the actual block, it is possible to compute the funds of each address balance, associated with each participant in the blockchain (ZOHAR, 2015).

To better understand the concepts behind a transaction, consider a simple example with two commercial partners, Bob and Alice, Bob wishes to transfer 50 coin units to Alice. To make the transaction between Bob and Alice possible, there must be means for (a) ensuring that Alice receives the coins; (b) registering and keeping the transfer in a transaction record; and (c) securing all information from frauds, including by Bob. In a traditional system, (a)-(c) are dealt with by a trusted entity.

The reason for Bitcoin's existence comes from the necessity of an electronic cash solution to allow digital payment transactions from one party to another without relying on a third party, in this case, a financial institution.

Removing the trusted entity leads to the need to build a trusted infrastructure operated by non-trusted entities, which is what blockchain is designed for. The idea is to distribute transaction information to nodes across a communication network and, using a suitable protocol, validate all transactions at network nodes. In general, Bob and Alice have their unique addresses associated with public cryptography keys through which transactions are parameterized. Transactions occur differently for distinct blockchains, though. The illustrative example of Figure 2.1 will be explained for Bitcoin transfer funds.

2.2.1 Transactions in Bitcoin blockchain

To perform a money transfer between Bob and Alice in a Bitcoin without a trusted central authority, like a bank, Bob needs to identify Alice in the network. Each user that participates in a blockchain receives a wallet address, which is the nearest concept to an “account". The wallet address corresponds to a hash of a cryptographic public key portion in the blockchain. In this case, Alice’s wallet address can be publicly discovered without any concerns. Alice also has her private key, which, as the name suggests, is secret and each user should keep it safe. It is with the private key that Bob signs its transaction coin transfer to Alice. As can be noticed, the private key serves to digitally sign[1] a transaction and to prove the ownership of the transaction. Once the transaction is digitally signed, it receives an identification hash, and it is transmitted to some blockchain nodes in the peer-to-peer (P2P) network. Each node verifies the transaction signature and continues forwarding the transaction to other peers until the entire network of nodes receives it. The digital signature also guarantees non-repudiation of a transaction as the recipient of the transaction cannot deny that the transaction was sent by the sender as the sender has the private key.

Bob’s digital signature, now included in the transaction, also guarantees data integrity. Indeed, as all transaction content is hashed, any modification of any information of the transaction would modify the transaction hash. The transaction is thus immutable. Furthermore, this structure helps to prevent the utilization of the same transaction more than once in the blockchain. The issue of spending the same transaction more than once is a known problem that can occur in monetary bills but is very unlikely in blockchain (ZOHAR, 2015).

For performing Bob’s transaction it is also necessary to check whether Bob has enough balance. The manner Bitcoin blockchain manages address balance is quite different from other blockchains. It is based on the concept of called Unspent Transaction Output (UTXO), which represents all unspent BTC amounts from a specific Bitcoin wallet (LI et al., 2022; ZOHAR, 2015). Hence, to perform Bob’s transfer to Alice the Bitcoin protocol initiates a balance verification searching for the remaining BTC amount on each unspent output from Bob’s previous completed transactions. Once enough amount is found then the new transaction is created pointing the unspent output(s) from previous transactions to the new inputs on the new transaction, as illustrated in Figure 2.1. Figure 2.1 shows two boxes representing transactions carried out by Bob. As can be seen, the leftmost box represents a transaction according to which Bob spent 20 BTC and left 40 BTC unspent. When he transferred 50 BTC to Alice, the new transaction used this amount of 40 BTC and other funds from other UTXO. Observe that the total amount involved

[1] Elliptic Curve Digital Sign Algorithm (ECDSA)

in this transaction is 52 BTC (instead of 50). The extra 2 BTC represents the necessary fee to include the transaction in a block by a miner (the exact value of the fee varies and is not relevant to the explanation). A transaction can have several inputs and outputs. However, one output is reserved to transfer the amount to the receiver. The rest is utilized to store the remaining BTC of the transfer transaction as shown in Figure 2.1 plus the fees. (ZOHAR, 2015) informs that once a transaction output is used, then all the BTC amount associated with it is considered spent. As a result, another transaction that attempts to access the already utilized output is rejected by the protocol.

The transaction fee is fundamental for making progress in the network as the fee represents the reward to miners. Until some miner does not include the transaction in a block, Bob's transfer does not take place. To do so, miners need to find a number (called nonce) for the block being validated. The nonce is also a field of the block header that is adjusted by miners in an attempt to create a valid hash for the block. The challenge is that when block header fields are hashed they correspond equally or less than the target (represented by the difficulty of finding a block hash with a specific number of leading zeroes) to be accepted by the network protocol (BITCOIN, 2021). The first miner who solves this problem takes the fee as payment. Through a distributed consensus protocol, the winner miner is recognized as the first to validate the block by the other nodes in the P2P network.

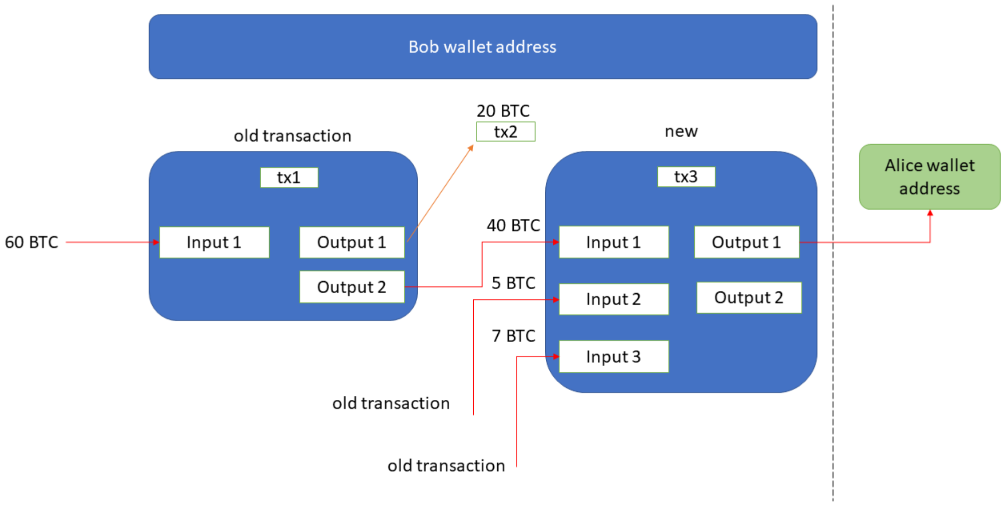


Figure 2.1 UTXO Unspent Transaction Output (ZOHAR, 2015; LI et al., 2022)

2.2.2 Transactions in Lisk blockchain

Differently from Bitcoin, in Lisk, there is no concept of UTXO. Instead, Alice and Bob have Lisk digital wallets that contain their Lisk balances. To Alice send 20000 LSK to Bob's wallet, she must inform a few parameters. For example, her passphrase, which corresponds to 12 words, the recipient address, Bob's digital address, and the transaction type of transfer transaction, as in Lisk there is more than one type of transaction, and some other information should be provided to sign the new transaction. The passphrase is unique and personal, and the digital signature is based on the famous Edwards Curves 25519 (DANIEL et al., 2017) algorithm utilized for cryptography purposes in many other blockchains and also in What's App applications. During this phase, there is not any balance verification. After the transaction is signed it should be sent to the Lisk nodes that will receive the transaction and include it in a transaction pool which is a queue that contains all transactions received by the node. A Lisk account forger during its time slot will forge a new block containing the validated transactions, sign the block, and send it to the network for validation by the Lisk protocol.

Upon Figure 2.2 it is possible to observe a straight transference of 20000LSK funds from Alice's wallet address on the left side to the Lisk wallet address on the right side.

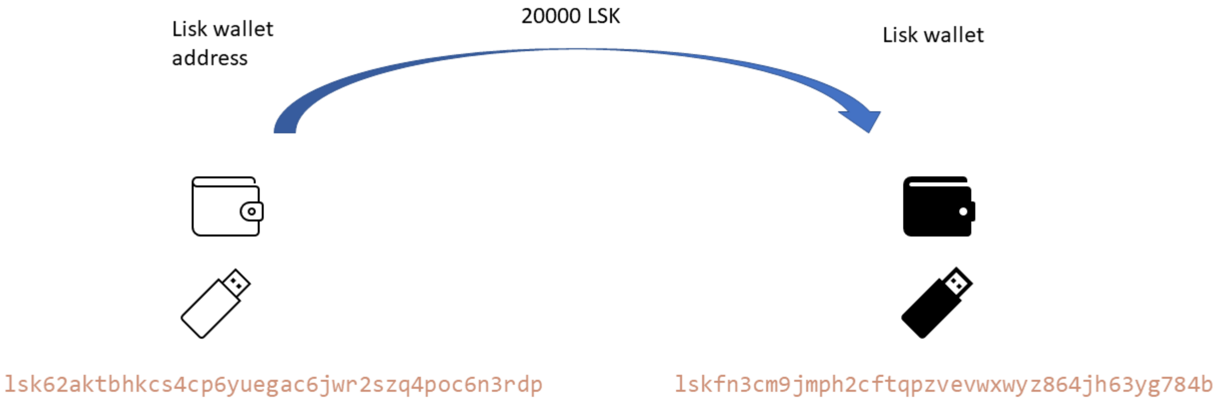

Lisk wallet address
20000 LSK
Lisk wallet
lsk62aktbhkcs4cp6yuegac6jwr2szq4poc6n3rdp
lskfn3cm9jmph2cftqpzvevwxwyz864jh63yg784b

Figure 2.2 Lisk wallets transference of funds

2.2.3 Blockchain properties and types

Despite being complex, blockchain is a technology with several benefits. Such benefits have been widely used in several fields including voting, supply chain, healthcare, IoT, and other applications (LIU et al., 2018). Here are some benefits usually common in distinct blockchains:

- transparency: Anyone can join the public network of Bitcoin and audit any transaction. When a new node joins a blockchain it requests the missing blocks to nodes synchronized with the actual version of the blockchain in an attempt to become a synchronized node too

- anonymity: pseudo users utilize accounts that do not contain a name, but a hash of cryptographic public keys (wallet addresses)

- auditability: the wallet address registers any transaction signed with its private key and anyone who participates in the network can audit the transaction. Also, those who do not participate in the network can utilize external tools to analyze a transaction. (ZOHAR, 2015) complements the information defining that is easier to follow the money and see where it is being moved in blockchain as it is publicly available to consult and this helps in analyzing coins that were considered to have been involved in illegal activity as each transaction performed leaves a track no matter how many times they change hands. For example, exchanges can refuse to accept them

- privacy: There is nothing that impedes a user from having several addresses for each new transaction, as the address can be easily generated

- tamper-proof: transactions are digitally signed and each block has a cryptographic hash of its block predecessor. Any attempt to change a piece of information in old data is identified by the network validators of the ledger

There are several blockchains currently available. However, the properties of blockchain are usually common, for example, the properties of decentralization and immutability are principal factors driving the adoption of blockchain technology because they allow blockchain-based applications to run without a centralized governing authority or trust between system users (MOORE, 2020).

- Immutability. The ledger is cryptographically secured, it can only be appended, and is immutable, also it is sorted chronologically

- Decentralization. No dependency on centralized governing authority nor trust between system users. Several nodes communicate messages between themselves, agree, and grow the same blockchain

- Fault Tolerance. Even in the presence of faults, other nodes can continue communicating and growing, eventually, the blockchain.

The kind of structure shown in Figure 2.3 allows its utilization in different types of blockchain platforms:

- Public permissionless blockchain is a type of blockchain in which anyone can run a node and join the network. When a new node arrives it requests the missing blocks on its node to other blockchain nodes that are already synchronized. These are the characteristics (PAHL; IOINI; HELMER, 2018):

  - Network is open to everyone
  - All nodes have the right to participate in the consensus protocol
  - Full ledger of transactions is accessible by anyone;

  Example: Bitcoin, Ethereum

- Public permissioned blockchains provide a hybrid model between the open environment of public blockchains and the private environment of private blockchains. These are the characteristics (PAHL; IOINI; HELMER, 2018):

  - Network access is controlled by a preselected set of nodes
  - A Preselected set of nodes controls the consensus protocol
  - Readability of the ledger can be public or private;

  Examples: Hyperledger Fabric, Ripple, Lisk

- Private blockchains, in this type of blockchain the participants are added and validated by a central organization. These are the characteristics (PAHL; IOINI; HELMER, 2018):

  - Network access is controlled by a single organization
  - Single organization controls the consensus protocol
  - Readability of the ledger requires access authorization

  Example: Multichain

In a blockchain, each miner and forger capable of including a block is rewarded with cryptocurrency. The reward mechanism from one blockchain to another can vary but on Bitcoin, the rewards are paid in BTC, and in Lisk are paid in LSK. The reward mechanism is important to maintain the whole blockchain and maintain validators.

As a curiosity, the first virtual currency was not Bitcoin, in fact, Wei Dai described the fundamental properties of a cryptocurrency in a famous paper called B-Money, however, he failed to find a consensus mechanism that also could prevent double spending and therefore allow the utilization in real-world implementation. Only in 1992, Dwork and Naor proposed combating email spam and Denial-of-Service (DoS) attacks with the idea of allowing system access conditional upon providing proof that a computationally expensive math problem had been successfully solved (MOORE, 2020).

2.3 BLOCKCHAIN INFRA-STRUCTURE

This section explains the blockchain infrastructure as Blocks, Merkle Tree, P2P, consensus mechanisms, and cryptographic hash functions. For the sake of explanation, it will be utilized the Bitcoin blockchain in examples. The explanation of Figure 2.3 has a demonstration of a chain of blocks in the Bitcoin blockchain. Each block is composed of a header and a list of transactions contained inside of it. The header contains several fields, one of which is the hash of the previous block, which makes the whole chain connected in chronological order through the hash of the previous block providing a secure data structure. The header of a block in a blockchain can vary from one blockchain type to another blockchain type, however, for generic purposes, it is possible to say that a header of a block contains several fields, and one of them is the hash of the transactions list contained inside of it, which is itself hierarchically composed of hashing the transactions inside of the block. Also, the hashing of each block takes the hash of the previous block and the hash of the block header fields, then the whole blockchain provides a very strong security infrastructure (ZOHAR, 2015).

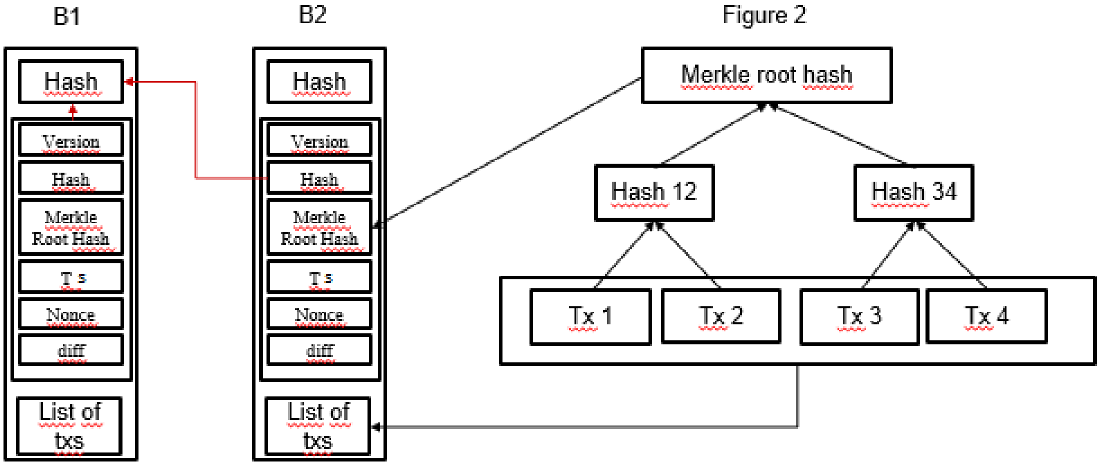


Figure 2.3 Bitcoin blocks based on (MOORE, 2020)

Agreeing on the version of a ledger is difficult and a challenge that the blockchain protocol solves. A consensus mechanism allows nodes to agree on the same version of the blockchain (ZOHAR, 2015). The peer-to-peer (P2P) network is utilized to establish communications between nodes, to allow blockchain scale as each node has a copy of the ledger allowing other nodes to communicate with it and process new blocks and transactions, also it is utilized to provide resilience as the network relies on each honest node connection to communicate transactions and blocks well, and without it, blockchain would be susceptible to attacks as such double-spending (ZOHAR, 2015). Each blockchain can have a different consensus mechanism and some will be presented in Section 2.3.2. For example, the procedure of mining a block is complex and very competitive as there are thousands of miners attempting to receive a reward for the mining. The blockchain network contains thousands of computers and each one of them tries to mine a block, hence, several computers can create a block approximately at the same time. In this specific scenario, the new block created by these nodes will point to the same parental block creating a fork in the blockchain. Such a block can contain conflicting transactions and also it can have a different version of the transaction log as the order of transactions inside a block matters in Bitcoin. It is exactly at this moment that the consensus mechanism of Bitcoin has two main rules to handle the problem, the first rule is the Proof-of-Work (PoW) that defines the block creation. PoW requires that a computationally hard math problem be solved by the computer that is running the Bitcoin program and wants to create a block. Verifying the solution of the problem is easy but solving it is complex as many guesses are necessary to be made with the goal of generating a unique and valid block identifier. The guesses are basically attempts to apply cryptographic hashing to some of the block field values. The second rule in the protocol running in the nodes is to solve the conflicting blocks at the same height by adopting the longest chain in the network and abandoning blocks in the shorter version of the blockchain. Together these rules bring consensus to the blockchain allowing every node to store the same history of blocks and transactions.

Many foundations of blockchain came from the past as the discovery of Haber and Stornetta (BAYER; HABER; STORNETTA, 1999) during the 1990s when they pioneered mechanisms for linked timestamping of digital resources by the utilization of linear hash chains to associate resources in chronological order and create a secure authenticated data structure that functioned as an unalterable digital timestamp ledger (MOORE, 2020). Such a concept is present in blockchain as each block is linked to the previous block by a cryptographic hash. In Haber and Stornetta's work (BAYER; HABER; STORNETTA, 1999) they explained that it is necessary to report events that could not have been predicted before they happened in order to establish that a document was created after a given moment in time. Hence, to establish that a document was created before a given moment in time, it is necessary to cause an event based on the document, which can be observed by others. Because of this necessity, the cryptographic hash functions can be used both to report events succinctly and to cause events based on documents without revealing their contents.

2.3.1 Data structures

In a blockchain such as Bitcoin, individual transactions within the block are referenced by a Merkle root that contains the hashes of all transactions in the block. To organize all data of blockchain it was introduced the concept of Merkle Trees (MERKLE, 1979), a key innovation that allowed timestamp data structures to be verified more efficiently and to be stored smaller. Merkle trees, which were introduced by Ralph Merkle in his 1979 doctoral dissertation and later patented in 1982, are a hash tree structure in which every terminal leaf contains the hash of a data block, and every parent leaf contains a hash of all of its child leaf labels (MOORE, 2020).

Merkle trees are projected for a leaf value that can be verified in relation to the root value known publicly, for that is necessary only the informed referenced pair values from the leaf until the root. It is really important in a blockchain, as it is used to resume efficiently the transactions contained in a block. For that is necessary to produce $2 * \log_2 N$ hashes (FERREIRA et al., 2017).

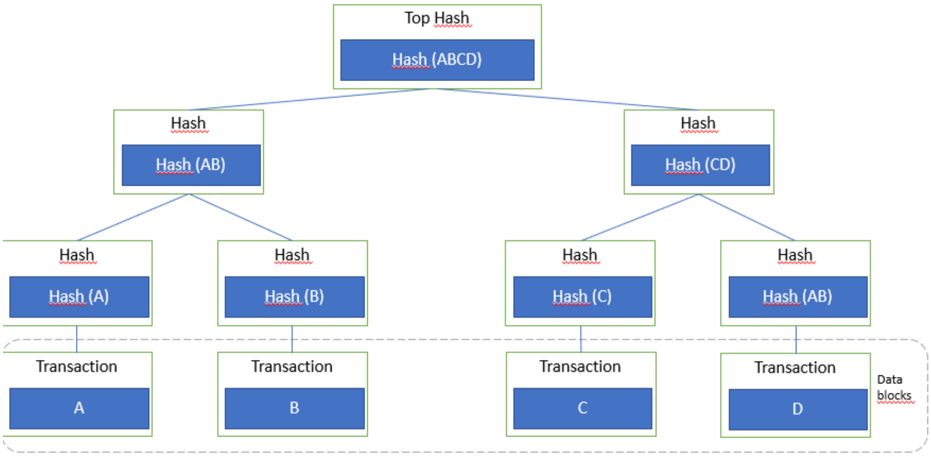


Figure 2.4 Merkle Tree (MOORE, 2020)

2.3.2 Consensus protocols

To organize the growth of a blockchain is necessary a protocol that allows the participants of the network to agree on a version of blocks in the blockchain. Such protocol is important to guarantee the integrity and consistency of transactions and their related data and it is known as a consensus mechanism. Also, it serves to validate the transactions in a blockchain. (PAHL; IOINI; HELMER, 2018) states that a consensus protocol allows all participants, and all copies of the blockchain, to agree on a single version of the true state of the network without the need for a trusted third party.

Following are the three most common consensus algorithms, after them will be explained the Delegated Proof of Stake (DPoS) present in the Lisk blockchain:

- Proof-of-Work (PoW). For a long time, it was considered the most utilized distributed consensus mechanism in blockchains. In PoW blockchain nodes have to provide a solution to a difficult math problem in order to have the right to append a new block in the blockchain. Such mathematical proof is crucial for determining the identification of a new block as well as in determining the winner of the race for solving the mathematical problem who is also called a miner. Such miner nodes compete trying to solve as fast as possible and demonstrate their PoW as only the fastest to solve it can include a new block in the blockchain. The reward mechanism for Bitcoin is in the form of cryptocurrency that is given to the winner of the mining race, actually, very stimulating and, therefore stimulates the participation of users. Transactions and blocks utilize a timestamp property that helps in preventing double-spending in the network as each newly appended block has to reference the preceding block by its unique identification based on a cryptographic hash function forming a linked timestamp distributed data structure. Then the block is broadcasted to other nodes in the system that validate it and if successful then add it to their copy of the distributed data structure (MOORE, 2020).

PoW is the base consensus protocol from the most known blockchains like Bitcoin and Ethereum. Such blockchains have a high number of nodes trying to solve a challenging puzzle. However, in such blockchains, several nodes try to solve such a challenge but only one node succeeds in mining a block to receive a reward per round, all the other node's efforts are completely wasted. Such characteristics of PoW brought attention inclusive in environmental concern as it is a waste of energy from all the nodes that don't mine a block. Such an issue has inspired the research of alternative consensus mechanisms that would not waste all the energy utilized by the participants of the network in an attempt to reduce the environmental impact.

- Proof-of-Stake (PoS). This distributed consensus mechanism has a lower computational cost than PoW. During the writing of this work, at the end of 2022, Ethereum performed a full switchover from PoW to PoS. In this type of distributed consensus mechanism the account that provides a higher stake of its cryptocurrency and dedicates a portion of it acquires mine capacity for a proportional opportunity to add new blocks to the blockchain. It means that more higher the stakes in an account higher the probability of mining a block, usually, an account with double of stakes than another account will have double the frequency probabilities for mining a block (MOORE, 2020).

- Practical Byzantine Fault Tolerance (PBFT). This distributed consensus mechanism also avoids the high computational cost of PoW. It was introduced by Liskov and Castro in 1999 and it is optimized to achieve consensus very fast, however, for that, it establishes a boundary for a number of malicious nodes in the system, usually, one-third of the number of the nodes in the network is the threshold (MOORE, 2020). Four phases define this consensus mechanism:

  - Phase 1: A proposed block is submitted to a specific validator node called the primary node in the network
  - Phase 2: Such primary node receives the block and broadcasts a validation request to the entire network
  - Phase 3: All nodes that received the validation request, validate it and return their output to the primary node
  - Phase 4: The returned outputs are evaluated then the submitted block receives the accepted status if the number of responses does not surpass the threshold of the number of malicious nodes, therefore a number equal to or higher than two-thirds plus one is required for successful validation;

PBFT requires more communication to achieve all phases and it can become an overhead that can cause scaling issues. Therefore, PBFT is most commonly used for custom blockchain implementations where the number of nodes is small, rather than public. Examples of blockchain protocols that support PBFT are HydraChain, and Hyperledger Fabric (MOORE, 2020).

- Delegated Proof of Stake. It is a consensus composed of a numeric quantity of nodes configured with delegate accounts, unique accounts responsible to forge blocks and change the state of the network. In Lisk, each node configured with a delegate account has a determined slot of time to forge a block in the network. In DPoS the slice time to forge a block is the default for each active delegate account during each round. Each round is composed of 101 active delegates, it runs on a round-robin sort and each active delegate has 10 seconds to forge a block (ALVES, 2020). Differently from Proof-Of-Stake in DPoS, the delegate accounts should vote, each vote has a weight, the weight is proportional to the delegate account stake and only the top 101 + 2 active delegate accounts that accumulated a bigger stake in votes can forge a block in the network.

  DPoS represents an evolution from PoS and blockchains already adopted it such as EoS, Lisk, and Tendermint.

2.3.3 Issues related to security, performance and cost

Despite the large applicability of blockchain, some security precautions should be taken in addition to utilizing the technology especially when is considered the source of information, a blockchain node, and an application that aims to connect with a blockchain node. Blockchain nodes must be synchronized as much as possible with the latest data in the blockchain. A blockchain node that is not synchronized with the other nodes in the blockchain can be a point of attack in the chain. Also, a node that behaves in a Byzantine way may be utilized as a point of attack in the chain. In this subsection let's learn about some security blockchain concerns. In the paper (ALVES, 2020) the author states that each node on a public blockchain is a computational node that is exposed on the internet by default and, therefore, exposed to Denial-of-Service (DoS) or Distributed Denial-of-Service (DDoS) attacks (DoS/DDoS). However, this is not the only issue related to security in blockchains. (SAAD et al., 2019; VASEK; THORNTON; MOORE, 2014) provided great text regarding attacks and their consequences in blockchains. Such attacks were grouped by peer-to-peer (P2P) network, consensus mechanisms, smart contracts, and more.

(MOORE, 2020) explains that the adoption of blockchain technology has been noted for both positive and negative cybersecurity implications as potential adversarial nations can have a lot of influence on domestically deployed global blockchain networks when they possess a disproportionate share of network computing power. For example, it is a risk having a concentration of 50% of the mining nodes for both Bitcoin and Ethereum be located in a specific country such as China.

Disproportionate control of a blockchain network's computing power can enable attacks such as deanonymizing and censuring users as well as selectively dropping transactions. It can also allow a blockchain's performance to be undermined through the destabilization of network consensus (MOORE, 2020).

(MOORE, 2020) complements that cybersecurity risks not only have the potential to directly impact blockchain users but also downstream risk effects through a third-party service provider that relies on blockchain technology and experiences an attack, and users of its service could be affected as well.

Among several types of attacks in blockchain technology (SAAD et al., 2019), the most well-known attacks are 51% Takeover Attacks, Selfish Mining Attacks, and Eclipse Attacks, and they are described below:

- 51% Takeover Attack. It is an attack in which an adversary gains control of over 50% of the mining power in the blockchain network. Once the 50% threshold of the mining power is exceeded, there are several destructive things that the adversary is empowered to do (SAAD et al., 2019). The attacker can block undesired transactions from being verified and added to the blockchain, also, transactions can be reversed and enable double-spending, finally, the attacker can execute a fork of the blockchain and only add new blocks that the attacker approves.

- Selfish Mining Attack. A selfish mining attack happens when block proposers refuse to propose a block that does not have interesting transactions for them. They just prefer to include the transactions of their interest on a block and propose them. A group of selfish mining can create a fork in a blockchain.

- Eclipse Attacks. Eclipse attack is a man-in-the-middle attack on a blockchain network. It is carried out by an attacker who controls a sufficient number of IP addresses within the network. This way the adversary monopolizes all connections from and to the node eclipsing it. Eclipsing a victim node is most commonly achieved by maliciously filling the victim node's peer tables with IP addresses controlled by the attacker through repetitive connection attempts to the victim node. Once the peer tables are full of attacker IPs, if the victim node restarts, it will connect to only the attacker's IPs. Attackers can try to force restarts with other attacks or simply wait for a node to restart of its own volition. Once the eclipse is achieved, the attacker can then either exclude the victim node from participation in the network or feed the victim node misleading information. This can be carried out discretely so that the victim node is unaware it has been eclipsed (MOORE, 2020).

2.3.4 Consistency, Availability, Partition Tolerance (CAP) Theorem and the Blockchain Trilemma

Failures can happen in a distributed system. Also, network partition needs to be tolerated and when it happens there are two options: consistency or availability. When choosing consistency over availability, the system will return an error or a time-out if particular information cannot be guaranteed to be consistent because of network partitioning. However, when choosing availability over consistency, the system will return with the most recent available information, even if it is not the latest and consistent because of network partitioning. However, as already noticed in the absence of network failure – that is, when the distributed system is running normally – both availability and consistency can be satisfied (SIMON, 2000; WIKIPEDIA, 2021). The consequences of a problem that results in a partitioned network can slow down application execution of transactions, or a not synchronized node can even provide the wrong information for merchants regarding the true state of a blockchain.

The CAP Theorem as defined:

- Consistency: Each node provides the most recent state of the system
- Availability: Every request receives a (non-error) response, without the guarantee it contains the most recent write
- Partition tolerance: The system continues to run even when there are partitions in the network

Similarly, the blockchain trilemma describes that is very difficult to achieve with great level of scalability, decentralization, and security in a single-chain blockchain (LU; QI; CHEN, 2023). Throughput is a metric that defines the number of successful transactions in blockchain (DINH et al., 2017), so achieving high throughput in blockchain would lead to the necessity of more storage. For example, Bitcoin is more than 350GB in size and is considered a low throughput blockchain with only 7 transactions per second, therefore, it will become more unlike a new node arrival as new nodes would need to download the full blockchain, consequently, the decentralization will not have the same scalability rate. Also, high throughput can limit the validation of a block in a more decentralized blockchain network as several peers would need to return validation results in a specified consensus time and because of network latency, it could result in issues such as for example limited use of block capacity, miscoordination, misallocation of transactions in a block, stale blocks (LU; QI; CHEN, 2023; AKBARI et al., 2020).

2.4 BITCOIN BLOCKCHAIN: QUICK INFORMATION

The Proof-of-Work (PoW) consensus of Bitcoin as discussed earlier in this chapter is based on the additions of timestamps to the blockchain using a Hashcash-like protocol that incorporates a Proof of Work in which network participants find the correct nonce needed to match a provided SHA-256 hash. The transaction histories are stored in a Merkle tree data structure in each block's header. This approach effectively uses economic resources to limit the number of user identities and prevent both takeover of the system and double spending activity (MOORE, 2020). Also, the implementation of the protocol supports a non-Turing-complete scripting language called Script (KLOMP; BRACCIALI, 2018).

The cryptocurrency of Bitcoin is set apart from other existing forms of digital currency. Bitcoin guarantees that it does not rely on any organization as no government is in control of its operation. Furthermore, there is no central entity able to apply monetary policy, and its supply will never surpass 21 million bitcoins (ZOHAR, 2015). Furthermore, Bitcoin does not require a banking account to hold it, and Bitcoin can be traded for goods or services with vendors who accept it (RAJAN; CAVALIERE; PALLATHADKA, 2021).

The following Bitcoin subsections will bring important information regarding the Bitcoin blockchain protocol and should be read carefully.

2.4.1 Bitcoin block structure

It was already shown in Figure 2.3 that transaction payload and header are the significant parts of a Bitcoin block. The header metadata uniquely identifies the block, and the transaction payload contains all transactions included in the block. By default, the Bitcoin block size was limited to 1MB. However, the size of blocks has other important implications as, for the sake of explanation, large blocks take longer to transmit and propagate through the network, and because of that more conflicting blocks would be created (ZOHAR, 2015).

2.4.2 Bitcoin header hash

The Bitcoin header hash is the main identifier of a block and it is stored in a separate database from metadata (FERREIRA et al., 2017). The header contains the metadata, as a hash to the previous block, nonce that is an arbitrary value utilized to validate transactions, version of Bitcoin, timestamp, Merkle Root hash, and the target difficulty to validate the block.

2.4.3 Bitcoin previous block hash

This field indicates the link to the previous block and it is stored in the block header. This field organizes chronologically the blockchain and makes it very secure (FERREIRA et al., 2017).

2.4.4 Bitcoin Merkle tree more information

The Bitcoin Merkle tree is a binary hash tree with k-bits associated with each node of the tree to help in finding a transaction inside a block. It is already known that a Bitcoin block can have thousands of transactions therefore the Bitcoin Merkle tree was projected for verifying a leaf value about a public root value known publicly, for that is necessary to send values of corresponding pair path from the leaf until the root. Furthermore, the Bitcoin Merkle tree is utilized to efficiently resume the transactions contained in a block, for that it is necessary to produce $2 * log_2 N$ hashes. To build a tree is necessary to start from the leaf that contains the transaction hashes. Then the leaves are grouped by pairs and their hashes produce a parent node, consequently, the parent node is grouped in pairs performing the same process until no more pairs exist then generating a root node called Merkle Root, Figure 2.4. To prove that a transaction is present in a block is necessary to send the transaction path in the tree, and this path is formed by the hash of complementary pairs. Hence, it is possible to rapidly verify a transaction among thousands of transactions. Furthermore, to verify if a transaction is present in a block is not necessary to send the full block but the header of the block and the path until the transaction (FERREIRA et al., 2017).

2.4.5 Bitcoin block height

Height is the block level in the Bitcoin blockchain and in other blockchains in general. Each block is linear and chronologically included in the blockchain. The difference between the first block, genesis, and the last block defines the position of the block in the blockchain (FERREIRA et al., 2017).

2.4.6 Nonce Bitcoin

The Bitcoin nonce is a number or variable utilized with the target difficulty field to prove that a miner performed work and succeeded in finding a block header that fills the criteria established by the challenge. For example, the miner attempts with brutal force to iterate the nonce value until the block header hash achieves a specific target difficulty, it can be like the header hash initiates with a sequence of 3 zeros for example (FERREIRA et al., 2017).

2.4.7 Bitcoin Difficulty

The difficulty in the Bitcoin blockchain is related to a computing challenge to find a partial hash collision. As an example, a hash algorithm attempts to always generate a unique output for specific input and a miner on Bitcoin needs to find a hash that satisfies a partial hash collision. The mechanism utilized to generate the collision is the nonce. As the nonce is a member of a Bitcoin block header, when it changes then the block header hash changes too, this happens because each minimal change on the input of a hash function will generate a new different output. For example, when the difficulty is adjusted to 1 bit it's only necessary to find a hash that initiates with 1 zero and any other value for the rest of 255 bits, which means the difficulty is adjusted with 2^{255}. As much as it is reduced the possible value of the difficulty to find the more difficult the challenge becomes. Therefore, more computational resources are needed, as more time is utilized by a miner, and more energy is utilized to solve the proposed challenge of generating a partial hash collision. On Bitcoin, a target of 10 minutes was established for solving the cryptographic challenge and the inclusion of a new block. Finally, to maintain the target of 10 minutes of block generation despite the arrival of new nodes and more powerful computers arrives then the target difficulty is adjusted on each new 2016 block by each miner (FERREIRA et al., 2017).

2.4.8 Bitcoin Scalability

(ZOHAR, 2015) states that the Bitcoin protocol is highly wasteful. It is known in the literature that a high amount of effort is expanded in arbitrary proof-of-work computations. All relevant information is saved at all mining nodes, messages are essentially broadcast through the network, and verification is always repeated. For these reasons, it appears the system would not scale well. Bitcoin's block size has been artificially (and somewhat arbitrarily) limited to 1MB per block and the size of the transaction is something around 0.5Kb. However, Bitcoin generates a low rate of transactions per second.

2.4.9 SPV and Bitcoin nodes

Bitcoin has different types of nodes and different types of users. There are necessities that some users need just to transfer funds from one wallet to another without any necessity to mine or store the full blockchain. Because of that some of the contents of the blockchain can be safely erased. They can manage a much smaller portion of the data. As informed by (ZOHAR, 2015) the Simplified Protocol Verification clients (SPV), also known as light nodes, allow users to connect to the network and download only the information they require. Such nodes are light enough to run on mobile devices, reducing storage costs for small users. Only miners and others with specific needs run full nodes and need to hold a full copy of the blockchain.

2.5 LIGHTNING NETWORK OFF-CHAIN SOLUTION

The lightning network is a decentralized network whereby transactions are sent over a network of micropayment channels between parties whose transfer of values occurs off-blockchain, it exists as a network above the Bitcoin network, therefore it downloads all blocks from Bitcoin and never runs after the latest block of the Bitcoin network (POON; DRYJA, 2016). Lightning nodes are the participants of the lightning network and can only fully execute when the lightning node is fully synchronized with the Bitcoin network. The establishment of such channels of communication occurs when two lightning nodes establish a connection between themselves and any other lightning nodes, for that one node initiates the creation of a channel that is governed by established rules between the lightning nodes. The creation of a channel is the starting point of such a relationship specifying how much could be transferred into the channel, the direction of the channel, and the rules to which each party should agree or be enforced during the channel's life, also settling the payments are done when the channel is closed, this can be done by mutual agreement or also can be enforced based on established rules (WIKIPEDIA, 2019).

The Bitcoin network as well as the Lightning network have the possibility to utilize a test network and also the main network. This work utilized Bitcoin and Lightning in a test network for conducting experiments and evaluating such technology.

2.5.1 Storage capacity

Running a lightning node requires the storage capacity to store all Bitcoin blocks already included in the Bitcoin network in the lightning node. This happens because the lightning protocol requires all blocks from Bitcoin to synchronize as lightning node execution relies entirely on the current state of the Bitcoin network. In the test network, it was necessary at least 46 gigabytes of storage capacity.

2.5.2 Creating a Channel

By creating a channel is established how much can be commercialized into the channel. It is possible to perform several micropayments also called Lightning transactions inside the channel without sending them to the Bitcoin network. Once a channel is closed the latest version inside a single transaction is sent to the Bitcoin network containing the latest version of payments (POON; DRYJA, 2016). For creating a channel is necessary at least that a connection exists between the nodes, then one node that funds the channel informs the other node public key and the amount of value locked in the channel such procedure sends the funding transaction into the Bitcoin network.

2.5.3 States of a Lightning Channel

A channel in the lightning network can be established between participants when the lightning node is fully synchronized with the Bitcoin blocks and when at least two participants agree to connect and one of them initiates a channel. The first important channel state after a default channel creation between two nodes is the state “CHANNELD_AWAITING_LOCKIN": in this state, it is necessary to wait for the inclusion of new 3 blocks in the Testnet network for an update in the state of the channel to “CHANNELD_NORMAL". After that, the channel is closed and the following flow occurs: “CHANNELD_SHUTTING_DOWN" => “CLOSINGD_SIGEXCHANGE" => “CLOSINGD_COMPLETE" => “FUNDING_SPEND_SEEN" => “ONCHAIN". “The FUNDING_SPEND_SEEN" => “ONCHAIN" will happen once the transaction is seen by the nodes and then a block with the transaction is included in the Bitcoin blockchain. Regarding other types of state please consult (POON; DRYJA, 2016; LIGHTNING, 2023b).

2.5.4 Payment

Payments in lightning can be accomplished by establishing a direct channel between two Nodes that want to transfer funds, therefore it is necessary to inform only the amount that can be transferred in a channel and if both mutually agree with the rules of creation of the channel (LIGHTNING, 2023a).

Another way of payment that can occur is through the generation of an invoice and then paying such an invoice. In fact, generating invoices and paying invoices are micropayments and can occur several times in a channel. The feature of payment for an invoice will execute successfully when it is possible to send the payment to the destination in the Lightning network (LIGHTNING, 2023a). For the sake of explanation let’s consider that Node 1 created 1 invoice in the amount of 3000 millisatoshis, Listing 2.1. The Listing 2.2 below shows the our_amount_msat field in the receiver channel of Node 1 prior to the invoice getting paid. Then Node 2 paid the invoice created by Node 1. After that moment Node 1 will visualize the payment of the invoice on its receiver channel at the amount field as in Listing 2.3.

Listing 2.1 Invoice created by Node 1 got paid by Node 2

```
{
 "label": "bread!!",
 "bolt11":
 "lntb30n1pj88gwasp5tap0gj5dz4qlu0vyymx004spm6h44zphkd2g7vv4zd6
 upjwzc4gqpp52zz4g5j2fcuktmfwcxn2pv548p8ht9c4qrlvt30d6su3fjcr7v
 eqdquv3jhxcmjd9c8g6t0dcsxyun9v9jqxqzfvcqp2rzjqfcxsh9gr28y6ngph
 mk90q05ejfydpq89tjjc5rl36lfmtcv424hkff225qqqgcqqqqqqlgqqqqqg
 q2q9qyysgq8ha8qm43ad5pdcvzqht3m5yvr48mds37h5nr6m2gsqsaufzwr3w8
 nv54f8lxluyqpwxdf8e30u5y9yphwejhy5ur0umyxhuukca9ehcpm0e5j8",
 "payment_hash": "508554524a4e3965ed2ec1a6a0b295384f75971500fec
 5c5edd43914cb03f332",
 "msatoshi": 3000,
 "amount_msat": "3000msat",
 "status": "paid",
 "pay_index": 10,
 "msatoshi_received": 3000,
 "amount_received_msat": "3000msat",
 "paid_at": 1685299701,
 "payment_preimage": "1fd68ff7864602d66f804dfbc8a1f00918d2735d2
 eed41c43e76d3d048183e1c",
 "description": "description bread",
 "expires_at": 1685299977
}
```

Listing 2.2 Receiver channel of Node 1 before invoice payment

```
 "channels": [
  {
      "peer_id":
"0270685ca81a8e4d4d01beec5781f4cc924684072ae52c507f8ebe9daf0caaab7b",
      "connected": true,
      "state": "CHANNELD_NORMAL",
      "short_channel_id": "2435669x35x0",
      "channel_sat": 2037,
      "our_amount_msat": "2037000msat",
      "channel_total_sat": 100000,
      "amount_msat": "100000000msat",
      "funding_txid":
"c82aeb98e3b473c4b4a17f58bc6368a9e094716493375e43ff552c0b03505579",
      "funding_output": 0
  }
]
```

Listing 2.3 Node updated amount after invoice got paid by Node 2

```
"channels": [
  {
      "peer_id":
"0270685ca81a8e4d4d01beec5781f4cc924684072ae52c507f8ebe9daf0caaab7b",
      "connected": true,
      "state": "CHANNELD_NORMAL",
      "short_channel_id": "2435669x35x0",
      "channel_sat": 2040,
      "our_amount_msat": "2040000msat",
      "channel_total_sat": 100000,
      "amount_msat": "100000000msat",
      "funding_txid":
"c82aeb98e3b473c4b4a17f58bc6368a9e094716493375e43ff552c0b03505579",
      "funding_output": 0
   }
]
```

At the moment that the received channel of Node 2 gets closed then a transaction is sent into the Bitcoin blockchain representing the Unspent Transaction Output (UTXO) for a Node 2 address, such amount becomes available as on Listing 2.4:

Listing 2.4 UTXO of Node 1 after closing the receiver channel of micropayments paid by Node 2. The transaction id exists on Bitcoin Testnet

```
{
      "txid":
"bbbe939848977a58f75409e31d6c3cda965ae4729e2a5e2c604ae6414f533816",
      "output": 0,
      "value": 2040,
      "amount_msat": "2040000msat",
      "scriptpubkey": "001443dfd348895cea1bb50da877d540d421078b7ce3",
      "address": "tb1qg00axjyftn4phdgd4pma2sx5yyrckl8rk4xezn",
      "status": "confirmed",
      "blockheight": 2436348,
      "reserved": false
   }
```

2.5.5 Routing

Routing is an important feature in Lightning networks. For the sake of explanation let's assume that Node 3 wants to send some funds to Node 1, however, Node 1 has only a bidirectional channel with Node 2 and no channel with Node 3. In such a situation, it is necessary to find a path between Node 3 and Node 1 allowing the funds to arrive at the destination. For that the feature of rotating the payment from Node 3 to Node 1 between Nodes in the network is mandatory, then a channel between Node 3 and Node 2 should exist. The command 'pay' can find a path automatically (LIGHTNING, 2023a).

2.5.6 Withdrawing directly to a Bitcoin account

After having funds in a lightning node wallet it is possible to visualize its outputs, such outputs are the Unspent Transaction Output (UTXO) in Bitcoin, therefore it is possible to withdraw any value from it directly to a Bitcoin address without requiring the explicit creation of a channel. The *lightning − withdraw* command allows sending the BTC amount from the lightning address of a node to a valid Bitcoin address directly (LIGHTNING, 2023a). The funds that are available for withdrawal are represented by the outputs field in the Listing 2.5.

Listing 2.5 Output of a Node in the lightning network

```
"outputs": [
  {
     "txid":
"5172ff4b529b5aacd372f753f6bc6e26559231718afd6ecde815ae676820c88b",
     "output": 0,
     "value": 5405,
     "amount_msat": "5405000msat",
     "scriptpubkey": "00140afbb37a4bfc38a429add251719eb28747aabf28",
     "address": "tb1qptamx7jtlsu2g2dd6fghr84jsar640egq57r9z",
     "status": "confirmed",
     "blockheight": 2435750,
     "reserved": false
  }
]
```

2.5.7 Hashed timelock contract (HTLC)

The purpose of an Hashed timelock contract (HTLC) is to allow for a global state across multiple nodes via hashes. This global state is ensured by time commitments and time-based unencumbering of resources via disclosure of preimages (POON; DRYJA, 2016). An HTLC has the following properties:

1 The smart contract can be spent by revealing a “secret’ together with a valid signature of the recipient (LIGHTNING, 2023b)

2 The smart contract can be redeemed after a set amount of time together with a valid signature of the sender (LIGHTNING, 2023b)

3 For transferring funds through a Lightning channel, an HTLC also comes with a third condition: The smart contract can be spent immediately by someone presenting a revocation key (LIGHTNING, 2023b).

Such a type of contract allows explicitly specifying the expiration time of the contract. It means that if the nodes in the channel didn’t agree to close the channel mutually then the rules of the contract will take place and execute it. The HTLC is the reason for a mutual close of a channel or enforces the close of a channel by the rules specified on it.

2.6 LISK BLOCKCHAIN

The Lisk blockchain allows registering transactions in blocks identified by a cryptographic hash. Each block refers to the hash of the preceding block establishing a sorted link between the blocks. Furthermore, any node with access to the blockchain can read and discover the global state of the network. To accomplish the blockchain goal necessary 5 components of a consensus protocol, Block proposal that generates blocks and attaches essential generation proofs, Information propagation that disseminates blocks and transactions across the network, Block validation that checks blocks for generation proofs, and the transactions within, Block finalization that reaches consensus on certain blocks and Incentive mechanisms that encourages honest participants and drives the system to move forward (ALVES, 2020).

Lisk is a blockchain created for the development of decentralized applications requiring low or no costs. Applications can be created utilizing their own transaction types and having their own logic based on Lisk SDK. The customized logic resides on an exclusive blockchain called Lisk Sidechain. It will be discussed Lisk Sidechain later in this chapter.

2.6.1 Lisk Consensus mechanisms

The introduction of Lisk-Byzantine Fault Tolerance (BFT) as a new consensus mechanism brings more efficiency and reliability into the blockchain network (HACKFELD, 2019). It is a based consensus mechanism on the famous Paxos (LAMPORT et al., 2001). Also, it is already proved by other implementations based on Paxos that it improves aspects of efficiency and reliability, also it is more tolerant against Byzantine faults. There are some important properties on Lisk-BFT as Safety, Liveness, and Accountability that exist to allow the achievement of consensus and permit the growth of blocks in the chain.

- Safety: Two honest block proposers never decide on conflict blocks, i.e., blocks that are not contained in the same branch of the block tree
- Liveness: An honest block proposer eventually decides on a block at any height
- Accountability: A block proposer can detect if a Byzantine block proposer violates the consensus protocol and can identify the Byzantine block proposer (HACKFELD, 2019).

Lisk-BFT also utilizes a block tree to represent the organization of blocks growing in blockchain as on Bitcoin. (HACKFELD, 2019) Defines the block tree from the beginning which is a root vertex also called genesis block that has a unique direct path from any block to its root vertex. Each vertex represents a block and their content can be data or messages. In blockchain, every block has a corresponding height, and in the mentioned tree it can be represented by the number of edges from the block to the genesis block, for example, the Genesis block has a height of 0. For the sake of explanation, it was defined as B0 ancestor of B' if they are distinct and is a directed path from B' to the genesis block. Also, it is possible to define that B' is a descendant of B when B is an ancestor of B'. When there exists a direct edge from B' to B then B is a parent of B', and consequently B' is a child of B. If there is a directed edge from block B to block B', then B' is called the parent of B and B a child of B'. Also, a chain can be defined by the path from a block without a child until the genesis block, and the block without a child is called the tip or head of the chain. Two blocks are called conflicting if they are not contained in the same chain of the block tree. On (HACKFELD, 2019) also is defined in the General consensus algorithm framework, Chapter 3, and on the LISK-BFT consensus mechanism, Chapter 4, the block finalize rule which prevents a blockchain from being reverted to a point before a finalized block height in case of the occurrence of multiple chains. The finalization rule of a block is very important to track if the chain that is growing is the longest and correct chain.

2.6.2 Lisk Merkle tree

The Lisk Merkle tree is an authenticated data structure organized as a tree (ALESSANDRO, 2020). From the beginning, almost all blockchains utilize Merkle Tree to organize their data structure, representation in Figure 2.4. However, as Lisk allows interoperability, it was chosen to utilize the Lisk Sparse Merkle Tree. One specific characteristic of a sparse Markle tree is that it allows the inclusion of data anywhere, not depending on the sort of block in relation to the blockchain. This approach is utilized extensively in solutions of Interoperability. It is a fast structure to verify if a transaction is inside a block, as per its binary tree structure organized in hashes, therefore it provides quick validation in the Lisk blockchain.

(RICOTTONE, 2021) described a Merkle Tree as a tree that has a root node, also known as Merkle root, and below its leaf nodes. Per the rule, a parent node can have a maximum of two leaf nodes. Computing the hash of two leaf nodes generates the hash of the parent node and the action of doing it recursively until the top leaves the tree with a single node. Therefore, it is very good for appending new data, also for authenticating transactions in a block, and verifying the order of transactions. However, allowing the insertion of new data at any position of the tree is not very efficient and would require the re-computation of several node hashes, including branch node(node above leaf nodes) and Merkle root.

In opposite to it and stated by (RICOTTONE, 2021) a sparse Merkle Tree is an authenticated data structure of a key-value pairs map, and a map is a collection of key values. Each key-value corresponds to a leaf node of the tree. It allows four operations such as lookup, insert, update, and delete efficiently. The order of insertion of data blocks does not matter and the root of the tree only depends on the final state of the key-value map collection. This can be accomplished with the Sparse Merkle Tree hash function as the input is the key value map collection and the output is the root of the tree, Merkle Root. Following the operations of Sparse Merkle Tree:

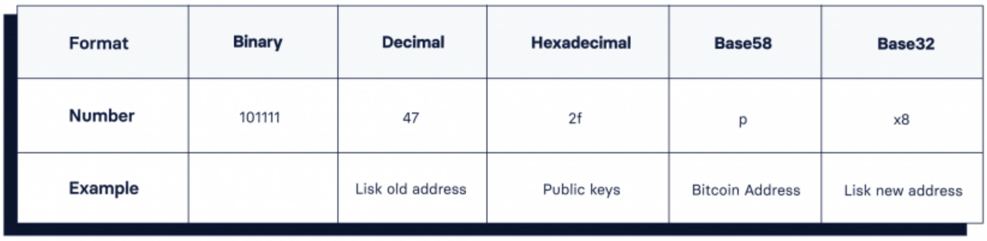
- Lookup: Returns the value associated with a key
- Insert: Inserts a certain key-value pair in the map collection
- Update: Updates value associated with a key
- Delete: Removes a certain key-value pair in the collection

2.6.3 Wallet account system

The new ID system, Figure 2.5, has a long address. It is composed of about 12 passphrases. (IKER, 2020b) for users of the Lisk platform, addresses will always be shown in a user-friendly Base32 format and include a checksum that detects typing errors.

The user-interface address representation is always 41 characters long, contains decimal digits and lower-case letters from the Latin alphabet and starts with the prefix "lsk". Basically, the encoding of the addresses for the user interface level is divided into three main steps listed below:

- Compute a 30-bit checksum of the 160-bit address and append this checksum to the address
- Encode the output into the Base32 format.
- Prepend the prefix "lsk" to the previous output.



Format	Binary	Decimal	Hexadecimal	Base58	Base32
Number	101111	47	2f	p	x8
Example		Lisk old address	Public keys	Bitcoin Address	Lisk new address

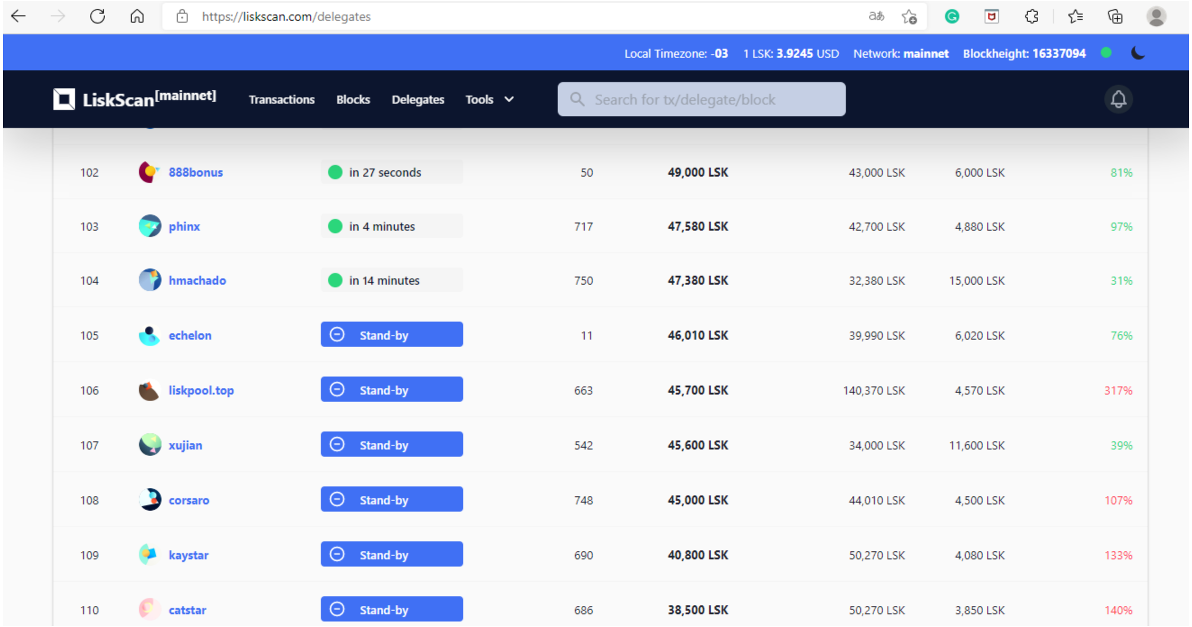
Figure 2.5 New Lisk address (IKER, 2020b)

2.6.4 Delegate accounts

Delegate accounts also known as block proposers are the unique accounts that can forge a block in the Lisk network. To forge a block each delegate account among the 101 top active delegate accounts and 2 standby delegates has a specific time slot of 10 seconds (LISK, 2021a). There is no competition between delegate accounts during a time slot, each time slot belongs to a single delegate account and only that specific delegate account can forge a block at that moment (ALVES, 2021a).

2.6.5 Voting mechanism

The voting mechanism is crucial in Delegated Proof of Stake (DPoS) as it is the base to define the top most voted delegate accounts to forge a block. Currently, to become a top 101 + 2 delegate is necessary to perform at least one self-vote on a delegate account. The self-vote is a base boundary to the voting capacity of the delegate account. Also, when any vote is cast the LSK value is locked and the account cannot utilize it for a transfer transaction. The vote limit capacity at the moment of writing this paper was 10x the weight of a self-vote. It can be expressed by this formula: $minimum(10 * selfVotes, totalVotes)$. For example, delegate A included a transaction for a self-vote of 200 LSK, which means that the maximum weight of vote that delegate A can have is 2000 LSK. Furthermore, even receiving more votes from other accounts the total weight cannot surpass 2000 LSK.



102	888bonus	● in 27 seconds	50	49,000 LSK	43,000 LSK	6,000 LSK	81%
103	phinx	● in 4 minutes	717	47,580 LSK	42,700 LSK	4,880 LSK	97%
104	hmachado	● in 14 minutes	750	47,380 LSK	32,380 LSK	15,000 LSK	31%
105	echelon	⊖ Stand-by	11	46,010 LSK	39,990 LSK	6,020 LSK	76%
106	liskpool.top	⊖ Stand-by	663	45,700 LSK	140,370 LSK	4,570 LSK	317%
107	xujian	⊖ Stand-by	542	45,600 LSK	34,000 LSK	11,600 LSK	39%
108	corsaro	⊖ Stand-by	748	45,000 LSK	44,010 LSK	4,500 LSK	107%
109	kaystar	⊖ Stand-by	690	40,800 LSK	50,270 LSK	4,080 LSK	133%
110	catstar	⊖ Stand-by	686	38,500 LSK	50,270 LSK	3,850 LSK	140%

Figure 2.6 Lisk scan explorer connected on mainnet network. The explanation of the column comes from the right to the left. The first column on the right represents the maximum capacity of weight. The second column represents the self-vote weight and the third column represents the total vote weight.

2.6.6 Transaction pool

Each Lisk node has a queue called the transaction pool. Such a queue has a capacity of 4096 transactions. Any transaction created and sent to a Lisk node is stored inside its transaction pool. Each delegate account during its forge time gets transactions from its transaction pool and includes them in a block (ALVES, 2021a). Transactions in the transaction pool are organized based on transaction fee, nonce from an account that signed a transaction, and a greedy algorithm is responsible for deciding for a transaction with a higher fee take the front of a transaction with a lower fee than arrived first in the transaction pool. The concept behind transaction priorities in the transaction pool is called dynamic fee and is explained in this chapter.

2.6.7 Block

A block is the data structure responsible for storing transactions in the payload property. While in the header property, it stores version, timestamp, height, previous block ID, generator public ID, reward, asset, and signature, represented in Figure 2.7 (LISK, 2021a). The default block size capacity depends on transaction size. (ALESSANDRO, 2020) informs that the capacity of a block is 128 transactions. (LISK, 2021a) informs that the default block payload size is limited to 15kb.

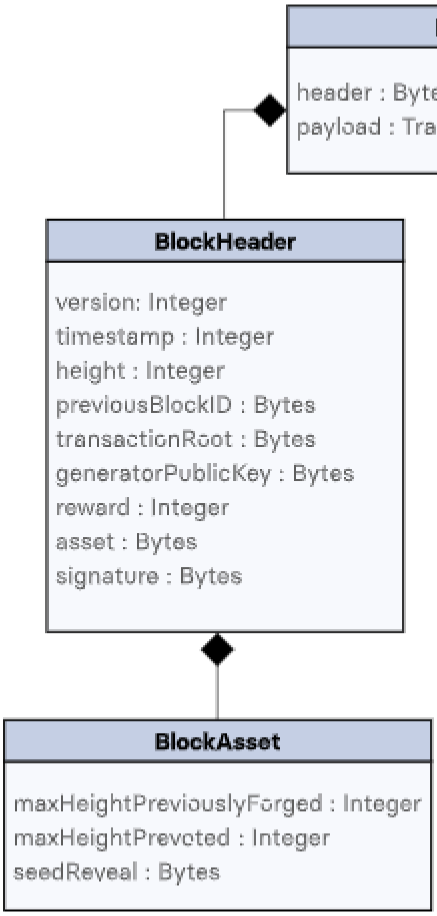


Figure 2.7 Block (LISK, 2021a)

2.6.8 Dynamic fee

Dynamic fee on version 5.x of SDK determines that a transaction fee from the transfer transaction type can accept different values. Originally, Lisk utilized a static fee system as its mechanism, SDK 2.x, in which there is a fixed fee per transaction type defined in the Lisk protocol. For example, a user had to pay 0.1 LSK to include a transfer transaction into a block, regardless of network congestion or block size usage. Despite its simplicity for users and developers, in a fixed-price mechanism, the resources are not distributed efficiently, and important limitations are imposed on the usability and scalability of the Lisk ecosystem. The dynamic minimum fee for a transaction depends on the type of the transaction and its size. Also, the transaction issuer has to be aware of this value if they don't want to send an invalid transaction. The Dynamic fee logic has a rule that gives priority to first executing the transactions with the highest fee. A greedy algorithm was implemented allowing delegates to first pick the transactions with the highest fee priority while respecting the order of the nonces per account (IKER, 2020a).

2.6.9 LSK cryptocurrency

LSK is the cryptocurrency of the Lisk blockchain. Different from Bitcoin, there is no limit on the number of Lisk created. Every new block forged by Lisk rewards a delegate with a new 1 LSK. At a glance, the reward was 5 LSK per each forged block, however, on every 3000000 blocks distance a halve happens and the reward decreases by 1 until it arrives on 1 LSK reward per block and remains this way forever.

2.6.10 Lisk Sidechain

Lisk sidechain is an exclusive blockchain that accepts custom transactions developed with the Lisk SDK libraries. Furthermore, a sidechain accepts only transactions supported by the Lisk blockchain network and custom transactions registered in the sidechain. This is a remarkably interesting characteristic especially because the sidechain does not allow transactions from different sidechains unless interoperability is implemented and supports cross-chain transactions. Furthermore, the transaction space on a sidechain block is reserved only for the custom transactions supported by itself and those supported by the Lisk blockchain. Each node in the sidechain network executes and accepts the same types of transaction (ALVES, 2021b). Compared with the Bitcoin network that accepts BTC transactions from different business needs, a sidechain can exclusively accept transactions from its own business. A sidechain allows scale in higher proportion than concentrating all transactions in a single blockchain as Bitcoin for example.

Upon a Lisk sidechain is possible to change some specific behavior from the regular Lisk mainnet/mainchain blockchain as block size, transaction pool capacity, block time or time each delegate account has to forge a block, number of delegates in the consensus mechanisms, network id, custom transaction fees per byte, reward after each new forged block. Also, it is possible to configure a white list of nodes that a node prefers to connect.

Upon a custom transaction is possible to specify a minimum fee amount to execute a transaction. Despite that on the Lisk SDK exists a rule that can be applied to calculate the fee amount based on the size in bytes of a transaction.

2.6.11 Lisk interoperability

(ALESSANDRO, 2022) states that Lisk interoperability is a solution that each chain that has a separate state and state transition function for blocks and transactions. Also, each chain has an independent set of validators that verify and finalize blocks. It means that each chain reorganizes itself independently when necessary. To do that Lisk utilizes the concept of Cross Chain Certificates (CCC), a data structure containing information about a specific chain. Let's utilize the example in Figure 2.8, the CCC has information of chain 2 signed by validators of chain 2 and submitted to chain 1. The submission process is done by relayers, and regular participants on the network. Certificates have several properties, among them are stated roots that authenticate the state transitions that happened on chain 2. Also, certificates contain the signatures of active validators of chain 2 and future validators of chain 2. Certificates allow the Lisk Mainchain network to trust information from the sidechain. Cross Chain Messages (CCM) are just information that corresponds to a transaction and is executed directly on the target chain. CCM are packed together with certificates into a Cross Chain Updates (CCU). The CCC has the following data structure:

- Block id: identify block header
- Height: used to validate a certificate
- Timestamp: used to validate a certificate
- State root: authenticates state of sending chain
- Validators rush: authenticates set of validators
- BLS signature: aggregate signature of current validators

In the above scenario is only possible to post a certificate that is newer than a previous one from the sending chain. The mainnet network is the central point of the Lisk ecosystem. It means that a sidechain that wants to participate in interoperability needs first to register on the mainnet to receive an ID and have a name. The sidechain registration transaction is composed by:

- Name of the sidechain
- ID of the genesis block of the sidechain
  - Used to compute the sidechain network ID
- Sidechain validators to sign the first CCU from sidechain

After sidechain registration on mainnet, then the mainnet is registered on the sidechain. In doing so, the sidechain's own chain ID and name are computed during sidechain registration. Mainchain validators sign first CCU from mainchain. All CCU are rooted by the mainnet.

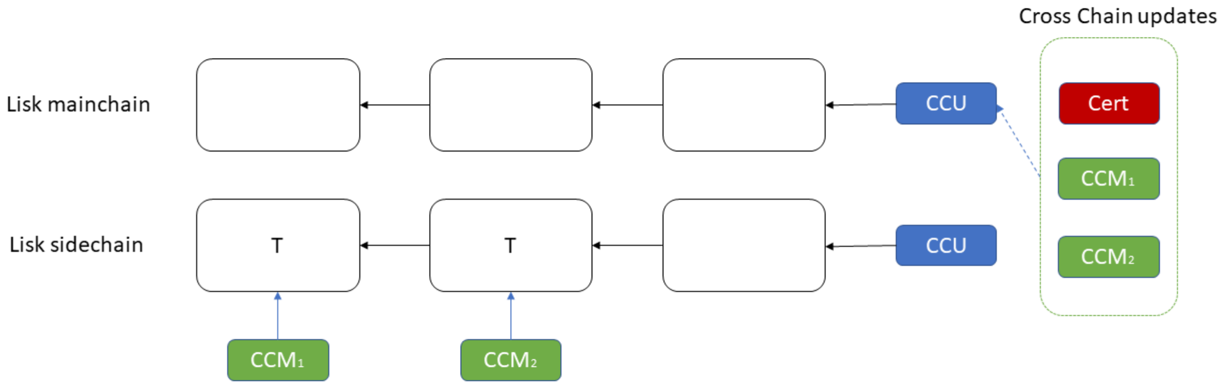


Figure 2.8 Lisk CCU. It contains certificate and CCM (ALESSANDRO, 2022)

2.7 THE TECHNOLOGY WE HAVE ADOPTED

The Lisk blockchain was chosen in this paperwork because of its characteristics including the possibility to create Sidechains and customize them, and also because of the capacity to develop custom transaction types and the utilization of the Javascript language, therefore, removing the necessity to learn a new language to the author, furthermore, it has the flexibility to define transaction fees, block processing time, size of a block, traceability, and more. Utilizing Javascript as the main implementation language does not require a learning curve by the author, and it allows us to concentrate only on the Software Development Kit (SDK) from the Lisk blockchain. Also, it makes it much simpler to integrate with the technologies from the Web. More details of this choice can be found in the next chapter.

2.8 FEW DIFFERENCES BETWEEN THE APPROACHES FROM LISK AND LIGHTNING NETWORK

Lisk is a blockchain technology that allows the development of a sidechain with an independent chain of blocks and transaction logic. Also, Lisk allows the customization of blockchain parameters such as block size, block interval, transaction fees, number of delegate accounts, peer connections, connectivity with external services and users, and much more. Another important feature is the ability to send data messages and store them on-chain differently from Bitcoin which does not allow it.

A lightning network is completely dependent on a blockchain network as it cannot execute alone. Lightning runs on top of Bitcoin and can only execute after downloading the entire Bitcoin blockchain on a Lightning node. It is an off-chain technology and it cannot customize the size of a Bitcoin block, Bitcoin block interval or change the cost of transaction fee. However, it can aggregate several micropayments in a channel deferring its payment until the closure of a channel. Also, it allows the establishment of several channels between peers and determines different rules per channel. Finally, after closing a channel is necessary to wait for the confirmation of the payment transaction in the Bitcoin network respecting the protocol. Alternatively, Lightning offers a withdrawal feature that does not rely on channels for processing payments.

2.9 INTERESTING PROPERTIES ANALYZED IN THIS WORK

The characteristics of blockchain/sidechain can solve the challenges of high fees and data privacy of current payment methods, however, the performance and scalability of the blockchain are very important properties that can drive the adoption of the technology. For example, (ALVES, 2021b) informed that requesting food online is common for millions of customers in the entire world, however, customer data privacy is still a concern to restaurants that receive customers' food delivery requests. DoorDash, a food delivery company reported a security leak of its customer-sensitive information, in such case more than 4.9 million data from customers, delivery workers, and merchants were hacked and leaked.

Security of user data and transaction performance is critical in retail operations. A behavior that can compromise user anonymity can impact customers' confidence in the retail service. Blockchain technology provides anonymity by design. On the same side, a retail transaction cannot take more than a few seconds. Transaction performance is a must in this type of application. This dissertation explores the use of blockchain technology services in retail operations. It proposes, builds, and validates an exclusive blockchain, also known as a sidechain, for restaurants as a proof of concept for using blockchain technology services in retail operations.

Would be blockchain a good payment method platform for enterprise solutions, specifically for retail items? To answer this question it was discussed blockchain technology, its reason for existence, and the requirements of a blockchain solution.

Following (LI et al., 2022) the mentioned properties are the main issues facing enterprise adoption of blockchain technology:

- Performance: All enterprise systems should be designed and built with an acceptable standard of performance as a minimum while taking into account problems such as scalability, latency, load, and resource utilization. Many factors could negatively impact performance, including high numbers of API calls, poor caching, and high-load third-party services. It's critical to ensure the end-user experience or integration of multi-systems across the entire ecosystem is not affected by any such issues

- Scalability: Scalability is the second big issue that needs to be addressed, as this is one of the core reasons why organizations still hesitate to adopt blockchains. The system must be able to accommodate ever-increasing volumes (number of users/devices/integrated applications, data, and throughput) over time, and is able to scale up and down quickly as the number of users changes drastically, as needed

- Security and Integrity: Requirements such as confidentiality, authentication, and integrity ensure that valuable (private and confidential) information is protected. Blockchain benefits primarily derive from the trust it fosters, its built-in privacy, security, and data integrity, and its transparency, as it incorporates a flow of data from complex mathematical operations that cannot be changed once created without being detected, and every transaction is encoded and connected, and therefore it is significantly more reliable than traditional journal methods. This unchangeable and incorruptible characteristic inherently makes blockchains safer and better protected against tampering and hacking of information
- Availability/Reliability/Resilience: The system must be available for use, and the downtime must be reduced to an acceptable level under any circumstances. For example, mechanisms to avoid single points of failure and adequate timeouts could be used to enhance system availability and reliability.

Chapter 3

RELATED WORKS

This chapter discusses the related papers with the proposed solution in this thesis including advantages and disadvantages in scalability, auditability, fees, cryptocurrencies, privacy-preserving, programming languages, and network utilization of each related work against the proposed solution. Also, it brings up performance evaluation in blockchains as throughput is a known issue in such technology.

In the context of the blockchain payment model, a paper relevant to the proposed blockchain for restaurants solution in this work was the scientific paper from (Kim; Kim, 2020). It proposes a simple payment model that uses basic cryptocurrency features, such as public key, private key, and digital signature, to eliminate the need for transaction intermediaries such as public key certificate and Payment Gateway (PG). The model can process e-commerce payments without registering an additional public key certificate, public key, or private key. The use of a digital signature guarantees the integrity and nonrepudiation of electronic payments, it eliminates the fees for intermediary services such as PG, thereby reducing the overall cost of operating e-commerce services.

The proposed solution from (Kim; Kim, 2020) utilizes the Multichain blockchain, a private blockchain that is compatible with the Bitcoin network. Multichain was created to solve some challenges in Bitcoin, however, it is a private blockchain and only its participants can audit transactions and blocks. The block size on Multichain is known as 32MB, and despite being configurable this is huge when compared with Bitcoin. However, developing smart contracts is not naively supported (PAHL; IOINI; HELMER, 2018). The consensus on Multichain does not rely on Proof-of-Work (PoW), it relies on a group of validators and only a single validator validates 1 block. It is a private blockchain so no need to generate an incentive for block creation, therefore no fee per transaction. Sending a message is not supported by default on Multichain and although it is possible to send data off-chain the storage does not occur at the blockchain level. Multichain was created focusing on financial institutions and not on public utilization (GREENSPAN, 2014).

(ARUNYADAV et al., 2018) discussed an Online Food Court Payment System using Blockchain Technology based on web and cellular applications developed over Ethereum smart contract. The customer can request their foods using a mobile app, a shopkeeper is responsible for completing the request in time and verifying the availability of ingredients. As it utilizes the Ethereum blockchain, it is possible to send messages on transactions. Also, Ethereum is a public blockchain and anyone can audit a transaction. The fees on Ethereum were higher at the moment of (ARUNYADAV et al., 2018) paper, however, in the second half of 2022 Ethereum updated its network to reduce the fees per transaction. Several thousands of transactions are included in the blockchain every day, also as per dynamic fee rules the greater the fee the faster the transaction will be included in a block. Transaction block size is dynamic, however, as there are several different smart contracts deployed in the network and also several regular transfer transactions executed on each block then a concurrence for a transaction space in a block is higher than executing similar business behavior in a private blockchain or sidechain. Despite the high concurrence, Ethereum offers compatibility with other crypto-assets based on Ethereum Request For Comments 20 (ERC20) (ETHEREUM, 2021).

(ESKANDARI; CLARK; HAMOU-LHADJ, 2016) discussed the utilization of real-world solutions based on Bitcoin as a Point of Sale in a cafe. It utilizes a framework for comparing Bitcoin Point of Sales solutions, scoring the competing systems on usability, deployability, privacy, and security. The solutions are based on low-volume transactions as a small business. In early Bitcoin Point of Sales designs, for each payment, the customer needs to wait 10 minutes on average for the transaction to be confirmed and included in the blockchain. However, the authors sidestep this issue by flagging the transactions as successful as soon as the transaction is broadcasted to the Bitcoin network, also known as a 0-confirmation transaction. They consider that it could work for a Point of Sale in a cafe as the amount of each transaction is small and it is not significantly riskier to take 0-confirmation transactions than to risk a credit card charge-back or even a customer leaving the store without paying. Cafeterias utilized a QR code in their terminals to inform customers of the exact amount of bitcoins that should be transferred to the restaurant's Bitcoin address and the customer could proceed to perform a transaction through a device.

Comparing the related works solutions with the necessities for this thesis solution it is possible to identify their limitations when exploring their online utilization. Bitcoin lacks a way to send data messages, and this way would not be possible to send the delivery address and store them in the blockchain. Also, the Bitcoin and Ethereum network generates greater transaction volume than other blockchains. Furthermore, the confirmation time of a transaction is still a problem for online utilization in Bitcoin. On Multichain it is not possible to audit in a public way the blockchain and the option to send data messages does not fulfill the requirement of this thesis solution which is to store sensitive information in the blockchain safely. The security of each compared blockchain in related works is still discussed in this chapter.

This thesis compares the context of the blockchain payment model proposed by (Kim; Kim, 2020), the Bitcoin Point of Sale proposed by (ESKANDARI; CLARK; HAMOU-LHADJ, 2016), the Online Food Court payment from (ARUNYADAV et al., 2018), and the Proof-Of-Concept (PoC) of Restaurants proposed in this thesis in the following categories:

- Solutions: The solution proposed in this thesis PoC of Restaurant sidechain, the solution Payment Model from (Kim; Kim, 2020), the Online Food Court payment (ARUNYADAV et al., 2018), the solution Bitcoin PoS from (ESKANDARI; CLARK; HAMOU-LHADJ, 2016)

- Send data message: Capacity to send data message when transferring funds between participants in a blockchain solution and the storage location of the data message

- Audit: Capacity to audit transactions and blocks

- Fees: Transaction fees to execute the transaction

- Cryptocurrency: Cryptocurrency utilized in the blockchain

- Scale: In the context of blockchains, scalability refers to different aspects: the throughput in the number of transactions per second, the types of transactions a system can process, and the interoperability with other systems (PAHL; IOINI; HELMER, 2018).

The results can be analyzed in Table 3.1.

Table 3.1 Solutions comparison

Solution	Send data message	Audit	Fees	Cryptocurrency	Scale
Lisk PoC Restaurant	Blockchain	Public	Very Low	LSK	High
Bitcoin PoS	N/A	Public	Vary	BTC	N/A Lightning
Ethereum FC	Blockchain HEX	Public	Vary	ETH/ERC20	Ok/slow
Multichain Payment	Off-chain	Private	N/A	N/A	Fast

Concerning cryptocurrency aspects, the Lisk sidechain can utilize the same cryptocurrency as the Mainnet of the Lisk blockchain. Also, several sidechains can be created with the proposed restaurant solution in this thesis increasing, even more, the scalability of the restaurant solution. Regarding the fees utilized in the new Lisk Restaurant sidechain, it is completely customized and therefore much cheaper than any solution proposed in Table 3.1. Following the Binance fee rate from Binance Exchange [1] the withdrawal on Bitcoin fee was around $15.87, 0.0004 BTC utilizing the Bitcoin network, and Ethereum fee $6, 0.002 ETH, utilizing Ethereum network. On Etherscan [2] is possible to visualize a

[1] https://www.binance.com Accessed on April 13, 2022

[2] https://etherscan.io/gastracker Accessed on April 13, 2022

fee of 61 GWEI [3] (ETHERSCAN, 2022a) equivalent to $3.92 for a transfer transaction on the Ethereum network, also such transfer would have an average of 3 minutes to be executed in the network, but, paying a higher GWEI fee allows a faster execution of it reducing the execution waiting time to 30 seconds or maybe 15 seconds. An Ethereum transfer to USDT [4] has a fee of $10.00. On Blockchain Explorer (BLOCKCHAIN, 2022) [5] is possible to verify each transaction fee on Bitcoin, and depending on the context it can be a high or cheap fee. These Bitcoin and Ethereum fees were incredibly high for a restaurant business and especially for a cafeteria as proposed on (ARUNYADAV et al., 2018). However, in the second half of 2022, the Ethereum network received a specific update called "The Merge" and its fees per transaction were reduced considerably returning to become an option for a restaurant business, therefore for another reference on gas fees please read on (ETHERSCAN, 2022b) and on Binance fees rate. Still utilizing the Binance Exchange fee rate an LSK withdrawal action has a fee cost of 0.1 LSK, or $0.2, and a regular transfer transaction on the LSK network has a fee cost of 0.000142 or $0.0002982. A sidechain food transfer transaction has only the sidechain fee, and because of that, it makes Lisk the cheapest network cost option for the proposed restaurant solution.

Ethereum has an interesting characteristic of the capacity of the number of transactions in a block. Each transaction required a specific amount of gas for its execution, hence the number of transactions in a block depends on the gas limit of the block and that was 21000 GWEI [6]. Let's say that all transactions have a fee cost of 50 GWEI and because of that the total number of transactions in a single block is $21000/50 = 420$. These 420 transactions can be executed in 15 seconds or more, and these transactions can be from different business needs, users, and companies as Ethereum has a single blockchain. Recently, and also because of an update called "The Merge" on the Ethereum network changed a while such explained logic regarding Gas limit per block and gas utilized. Regarding Bitcoin, it has a limit of 1MB block size to be mined in 10 minutes on average, and it can lead to more than 3000 transfer transactions in a single block on a single Bitcoin blockchain.

In the context of data privacy, important characteristics of the solution proposed in this paper were identified on the paper from (LIU et al., 2018) where it was discussed privacy-preserving data sharing for electronic medical records using blockchain technology, cloud, and Content Extraction Signature (CES). CES allows the users to remove sensitive portions from the original signed message and regenerate valid extraction signatures by themselves without extra interactions. Besides, it has the merits of low communication overhead, high efficiency, and privacy preservation. It is safe and the data is registered at the blockchain level.

The paper from (ALVES, 2021b) also is considered a reference for solving the problem of data privacy in the blockchain being referenced by the paper (FOLHA et al., 2022). It utilizes only the private key from the sender of a transaction and the public key of

[3] Ethereum gas unit price

[4] Tether stable coin

[5] https://www.blockchain.com/explorer?view=btc

[6] https://etherscan.io/gastracker Accessed on April 13, 2022

the recipient of the transaction to encrypt the sensitive information of the sender. Then only the recipient of the transaction can decrypt the client's sensitive information by the utilization of its private key, the nonce utilized in the encryption of the message, and the encrypted message in the transaction.

Comparing the Payment Model from (Kim; Kim, 2020) that utilizes Multichain private blockchain with (LIU et al., 2018), there is no native support for sending data messages in the blockchain but only off-chain, and therefore the storage is outside of blockchain adding extra complexity. Comparing Bitcoin Point of Sales from (ESKANDARI; CLARK; HAMOU-LHADJ, 2016) with (LIU et al., 2018) is also irrelevant since there is no native support in the Bitcoin blockchain for sending data messages between participants and having the messages stored in the blockchain. The proposed solution in this thesis PoC of Restaurants can send data messages between participants. Also, it can safely encrypt messages based on the recognized Edwards curves Ed25519 algorithm (DANIEL et al., 2017), it can include a data message inside a transaction and into the Lisk blockchain/sidechain, this is important to guarantee privacy between a customer that wants to buy food online and not have its sensitive information exposed without any security. Also, it is important for restaurants that want to sell food online and deliver the food to a customer. Furthermore, from the perspective of utilizing the blockchain to send news from restaurants to its customers or from sidechain owners to its restaurants, customers, or even between customers.

Comparing the solution from (Kim; Kim, 2020) and utilizing the analyses from (SAAD et al., 2019) paper it is easy to understand that a private blockchain offers less risk than a public blockchain, however, the payment model solution based on the Multichain blockchain does not offer other characteristics considered important for the restaurant solution proposed in this thesis and already mentioned in this chapter. The Bitcoin Point of Sale solution proposed by (ESKANDARI; CLARK; HAMOU-LHADJ, 2016) utilizes the Bitcoin network, and attacks against peer-to-peer (P2P) system and blockchain applications were already proved as stated in (SAAD et al., 2019) as well on smart contracts, the technology utilized on the Online Food Court Payment (ARUNYADAV et al., 2018). This thesis proposes the PoC of Restaurants based on the Lisk blockchain and an evaluation of the performance on the created sidechain for restaurants.

Regarding the evaluation of blockchain performance, the paper from (AKBARI et al., 2020) utilized a simulator developed by researchers at ETH for evaluating the Bitcoin blockchain and observing its behavior when blockchain parameters such as block size or block interval (also known as block time) were modified. Authors of (AKBARI et al., 2020) explained that is intuitive to try increasing throughput by increasing block size and shortening block interval (also known as block time), however doing so can lead to an increase in the number of stale blocks (blocks that are not included in the longest chain) in the blockchain, and that is an undesired behavior in blockchains. For example, Bitcoin by default has a block size of 1MB (it can be configured by a higher size) and it creates a new block every 10 minutes, but Ethereum, differently from Bitcoin, has a smaller block interval, it usually creates a new block on every 15 seconds but has a limited gas that can be utilized on each block. Attempts to increase transaction throughput already existed in blockchains but they lead to other problems, for example, the necessity of high storage for the ledger making it difficult for anyone to run a full node or audit the blockchain on an ordinary computer, or even the problem of the reduced number of nodes and therefore reduced decentralization to allow faster validation and execution of transactions, for example, HyperLedger Fabric (PAHL; IOINI; HELMER, 2018), or worst as security implications in the cause of the creation of undesired stale blocks (AKBARI et al., 2020)

3.1 PROGRAMMING LANGUAGE COMPARISON ON EACH BLOCKCHAIN

In the context of programming language comparison, Table 3.2 compares the programming languages utilized on each blockchain solution exposed in this chapter concerning adaptation to web applications such as those proposed in this work for allowing customers to buy food through a web browser. The Javascript language utilized by Lisk also was a determinant for blockchain platform choice as it was not necessary for learning a new programming language.

Table 3.2 Programming Languages comparison

Blockchain	Programming Language	Turing complete	Learning curve
Lisk	Javascript / typescript	Yes	None
Ethereum	Solidity	Yes	Yes
Bitcoin	Script	No, Turing Incomplete (No loops)	Yes
Multichain	Javascript	Yes	None

3.2 SIDECHAIN SUPPORT

In the context of sidechain support on the blockchains compared in this section, Lisk has support for sidechains, therefore it provides the opportunity to create custom transactions, change configurations, integrate with plugins, and enable it for public or private audiences. Bitcoin, Ethereum, and Multichain run on their own blockchains and this is the main reason for the limitation in leveraging the rate of executing transactions per second. On the other hand, Bitcoin has an off-chain solution called Lightning network that provides fast transaction processing, as Ethereum has, for example, Raiden network providing fast transaction processing, both are off-chain solutions (YANG et al., 2020).

3.3 PARALLEL BLOCKCHAIN AS PARALLEL SIDECHAIN OR MULTI CHAINS

It will be utilized the term multi-chain for parallel blockchain/sidechain and it doesn't have any relation with the blockchain called Multichain. The Lisk sidechain can run independently utilizing the same cryptocurrency as the main Lisk blockchain. For allowing interoperability then the unique rule is to send a transaction to the main Lisk blockchain once every 30 days. This is to inform you that the sidechain is alive. Knowing that then it is possible to increase the rate of execution of transactions per second by running several independent sidechains that do not need to communicate between them. In a sidechain, there is no competition between users' transactions from different chains.

3.4 MATRIX OF RELATED WORKS

The matrix of related works in Table A.1 includes works in several areas as the main focus of the problem stated in this dissertation. The matrix includes works regarding data privacy of user sensitive data utilizing a hybrid on-chain solution and off-chain solutions (utilizes external solution to store data, interacts with on-chain solution), for example in (ZYSKIND et al., 2015) it was defined a protocol for administration and access control authorization of user data which stores sensitive data off-chain and stores indexes linking such data on-chain; another hybrid solution can be found in (LIU et al., 2018) that utilizes the cloud and smart contracts to manage access control and user sensitive data storing on-chain only indexes of user data; however, a complete on-chain solution can be found in (ALVES, 2021b) work which utilizes public key, private key, nonce and a message with a cryptographic algorithm called Edwards Curves 25519 (DANIEL et al., 2017) to encrypt user sensitive data that only a recipient of a transaction can decrypt. Also, the matrix of related works contains works that performs performance evaluation as can be found in (AKBARI et al., 2020) that utilized a Bitcoin blockchain simulator to configure different parameters of block size and block interval with a goal to find a suitable configuration that could improve the performance and security in Bitcoin blockchain, furthermore it was observed security aspects in such evaluation; another work that evaluates performance but in this case in a sidechain solution is from (ALVES, 2021b) that performs an empirical study with experiments within a custom network with nodes; also there is the work from (FAN et al., 2020) that performs a systematic survey for evaluating blockchain performance of several blockchains and proposes the best suitable methods for optimize performance in blockchains dividing the methodology in two categories as empirical analysis and analytical modelling, then it describes the most common simulators utilized for performance evaluation in blockchains as Blockbench (DINH et al., 2017) a simulator for analyzing private blockchains, BlockSIM (ALHARBY; MOORSEL, 2020) a simulator for analyze PoW based blockchains, and also (FAN et al., 2020) suggests the utilization of simulators, especially, when it's expensive to reproduce experiments with real blockchain networks. However, despite such simulators trying to mimic a blockchain environment, they cannot be utilized in many consensus mechanisms because lack of support for these cases, therefore it was recommended to perform an empirical study technique for evaluation. However, when the subject is improving blockchain performance in off-chain networks then the matrix highlights the work from (KHAN; STATE, 2020) especially focusing on the Lightning Network, and Raiden Network, which can establish communication channels between sender user and recipient user to improve the throughput for executing transactions respectively for Bitcoin and Ethereum blockchains. Hence, Depending on each scenario each performance solution could not attend to the requirements proposed in this dissertation as fast transaction, customer-sensitive data preservation, scalability, or transaction fee low costs. Finally, the matrix contains works that describe blockchain main concepts as (ZOHAR, 2015; CHAUM, 1982), consensus mechanisms (NAKAMOTO, 2009; LAMPORT et al., 2001; HACKFELD, 2019; GREENSPAN, 2014), comparisons in blockchain protocols (XIAO et al., 2020; PAHL; IOINI; HELMER, 2018; LI et al., 2022), works related to cryptocurrency, transaction fee costs, taxation (RAJAN; CAV-

ALIERE; PALLATHADKA, 2021; CAMPBELL-VERDUYN, 2018; YERELI; SAHIN, 2018; SINKOVIC; PRIBISALIć, 2022; BLOCKCHAIN, 2022; ETHERSCAN, 2022a), and regarding the suggested case study works describing the application of blockchain in e-commerce scenario as (ARUNYADAV et al., 2018; ESKANDARI; CLARK; HAMOU-LHADJ, 2016; ALVES, 2021b; Kim; Kim, 2020).

The research technique based on keywords facilitated in finding relevant works in the blockchain performance evaluation field as shown in Table 3.3. The equivalent logic helped in finding suggested case study similar works as proposed in this dissertation as well as helped in finding works with the goal of reducing the fee, removing the middleman, and improving transaction processing time as informed in Table 3.4. During the research, one issue was finding several results in the supply chain area.

Table 3.3 Main search keywords throughput evaluation (RAJAN; CAVALIERE; PALLATHADKA, 2021)

Keywords	Search tool	Results	Reference
blockchain + throughput + parameters	Portal Periodicos	117	(AKBARI et al., 2020)
blockchain + throughput + sidechain	Portal Periodicos	80	(ALVES, 2021b)
blockchain + performance + evaluation + empirical analysis	Portal Periodicos	29	(FAN et al., 2020)

Table 3.4 Main search keywords blockchain use case restaurant (RAJAN; CAVALIERE; PALLATHADKA, 2021)

Keywords	Search tool	Results	Reference
BLOCKCHAIN CAFE FOOD PAYMENT BUY	Google scholar	6620	(ARUNYADAV et al., 2018; ESKANDARI; CLARK; HAMOU-LHADJ, 2016; ALVES, 2021b)
BLOCKCHAIN + E-COMMERCE	Portal Periodicos	647	(Kim; Kim, 2020)
BLOCKCHAIN + E-COMMERCE	Google Scholar	18000	(Kim; Kim, 2020)

3.5 SUMMARY

This chapter presented a comparison between related works and the proposed restaurant solution in this thesis. The comparison between Bitcoin, Ethereum, and Multichain blockchains against the Lisk blockchain utilized in this thesis identified important characteristics covering scalability, cryptocurrency, auditability, sending data messages, fees, network, privacy, and programming language. Based on the study presented it was decided to utilize the Lisk sidechain in the proposed solution for restaurants.

Chapter 4

PROTOTYPE DESIGN OF SIDECHAIN SOLUTION

This chapter explains the solution proposed in this thesis. The restaurant solution was created to allow customers and restaurants to utilize blockchain technology for requesting and paying food through an online blockchain platform offering fewer fees than regular payment models with a fast process rate. The proposed solution utilizes a single cryptocurrency around the world in a safer manner it protects the privacy of customers.

The Lisk restaurant solution offers and utilizes Javascript library classes allowing any restaurant to participate in the sidechain solution, consequently adding the ability to process payments with blockchain technology and store data in the blockchain. Such a solution utilizes a cryptographic elliptic curve algorithm for the encryption of customer-sensitive information, furthermore, the chosen algorithm utilized is based on assets that are available to the transaction owner, for example, its passphrase, and the restaurant public key. A detailed explanation will be provided in Section 4.3.

First, this chapter introduces the requirements of the new Proof-Of-Concept (PoC) of Restaurants utilizing the Lisk sidechain, then it describes the importance of Edwards Curves 25519 (DANIEL et al., 2017) in the solution as securing the privacy of sensitive data, then is provided a resume of the legacy solution of PoC of restaurants proposed in (ALVES, 2021b), and finally, the new solution is described.

4.1 REQUIREMENTS OF NEW POC LISKRESTARUANT SIDECHAIN

The proposed blockchain solution for restaurants of this thesis requires the following:

- Fast block generation on the blockchain, maximum of 10 seconds per block, configuration goal for 5 seconds of block interval
- More than 25 restaurant food transactions capacity in a single block
- Possibility to send data messages in a food transaction
- Store food transaction and its data in the blockchain inclusive the customer's sensitive data encrypted based on Edwards Curve 25519 (DANIEL et al., 2017)

- Low fees for each food transaction, maximum of 1% fee per transaction
- Not allow the inclusion of custom transactions with different business needs in the blockchain
- Public traceability to consult any transaction in the blockchain

4.2 FOOD REQUEST FLOW

This section describes the flow of requesting food in a restaurant that participates in the sidechain solution for restaurants. Figure 4.1 shows the flow of requesting food upon the first restaurant solution (ALVES, 2021b) and it is available for testing purposes in [1]:

1 A user can access the website of a restaurant that participates in the sidechain solution

2 A user chooses a food in the available menu of the restaurant

3 The restaurant then makes available options for the user to place a request, a QR code, or a link to inform the passphrase of the user's wallet address

4 After choosing the option the user places an order

5 The restaurant receives the request and it can prepare the food if it is a valid transaction

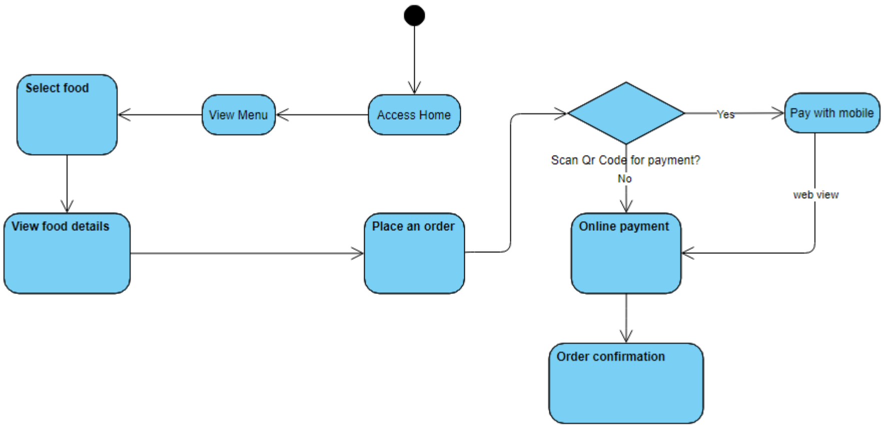


Figure 4.1 First restaurant solution food request flow(ALVES, 2021b)

[1] https://www.liskrestaurant.com

4.3 PRESERVING THE USER'S SENSITIVE DATA WITH EDWARDS CURVE 25519 AND EDWARDS-CURVE DIGITAL SIGNATURE ALGORITHM (ED-DSA)

The purpose of this Section is to describe the base of the cryptography utilized to protect customer-sensitive data included in a Food transaction. The preservation of the customer-sensitive data is performed on-chain, which means that the encrypted data will be available for audit. Though (DANIEL et al., 2017) defines the Curve25519 as an elliptic curve utilized in elliptic curve cryptography with 128 bits of security and 256 bits of key size, Edwards-curve Digital Signature Algorithm (EdDSA) that is a digital signature scheme, and Ed25519 that is an EdDSA signature scheme using SHA-512 and Curve25519. EdDSA is considered a discrete logarithm scheme that utilizes a nonce unique to each signature as it chooses the nonce deterministically utilizing the hash of part of a private key and the message signed. Therefore, there is no danger that a broken random number generator used to make a signature reveals the private key. For message encryption with a passphrase is necessary only the message encrypted, the passphrase of the user that is creating a transaction, and the public key of the recipient of the transaction as displayed on Listing 4.1.

Listing 4.1 Message encryption with algorithm based on Ed25519

```
encryptMessageWithPassphrase:(message:string, passphrase:string,
recipientPublicKey:Buffer) => EncryptedMessageWithNonce;
```

The result of the method encryptMessageWithPassphrase is an object that contains the ciphered message and the nonce utilized to encrypt the message. The ciphered message and the nonce are stored in the Food transaction and included in the blockchain that is public. As explained previously, even in the possession of the nonce utilized to cipher the message it is not possible to reveal the private key utilized in the encryption process thanks to Ed25519 and EdDSA. However, the nonce is unique and vital to decipher the message. For message decryption is necessary the ciphered message, the nonce received in the encryption method, the passphrase of the recipient, and the sender's public key, therefore only the recipient can decipher the message utilizing its passphrase, and it is guaranteed that the message was sent by the sender as its public key is required for message decryption as shown on Listing 4.2.

Listing 4.2 Message decryption with algorithm based on Ed25519

```
decryptMessageWithPassphrase: (cipherHex:string, nonce:string,
passphrase:string, senderPublicKey:Buffer) => string;
```

4.4 FIRST SOLUTION

This Section describes the first version of the PoC Lisk Restaurant (ALVES, 2021b) that utilizes Lisk Software Development Kit (SDK) 2.X, the main elements of its solution:

- Custom Transactions

- Web Application of each restaurant
- Backend application of each restaurant
- Sidechain nodes

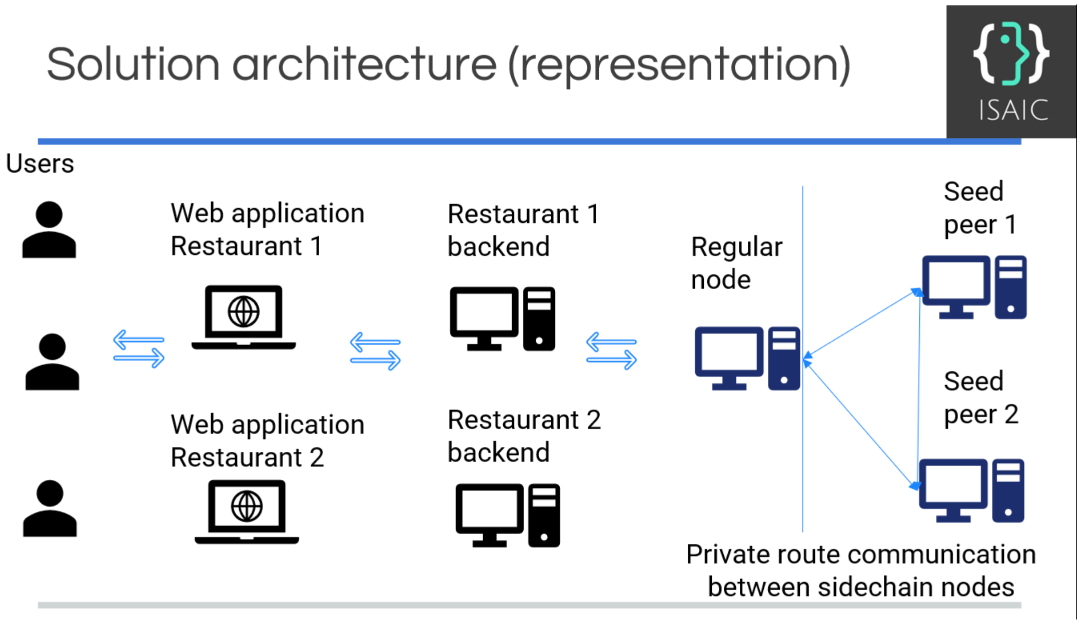
4.4.1 Custom transaction types

Custom transactions are the main feature of any sidechain network in the Lisk ecosystem. On the first restaurant sidechain solution, there are MenuTransaction, FoodTransaction, and RefundTransaction custom transactions. MenuTransaction allows the retrieval of an entire food menu option from a Lisk address belonging to a restaurant. This way any restaurant can have its food menu create a MenuTransaction and broadcasting it into the sidechain. The foodTransaction transaction type allows a customer to request food to a specific restaurant Lisk address respecting the privacy of sensitive information. The sensitive information of a food transaction is included encrypted in the transaction and broadcasted into the sidechain (ALVES, 2021b).

4.4.2 Web Application and customer sensitive data

A restaurant web application communicates with its restaurant backend API performing a public network request to its IP address. Upon Figure 4.2 it is possible to understand the interaction of restaurants that are connected to the restaurant's sidechain through a backend API. When a customer requests food on a restaurant web application, then a FoodTransaction is created on the client side and the sensitive information from the customer as a delivery address, phone number, and name is stored encrypted in the transaction. This information is stored encrypted in the blockchain using the customer passphrase and restaurant public key through a specific encryption method based on Edwards Curve 25519 (DANIEL et al., 2017).

The FoodTransaction allows the encryption of customer-sensitive information. However, to decrypt the sensitive information is necessary to utilize the public key from the sender address of the transaction, and the passphrase from the recipient of the transaction, which can be the customer and restaurant respectively for example. Therefore, a FoodTransaction stores customer-sensitive information that only a restaurant can decrypt, and other encrypted fields that only the customer can decrypt. When all information is included in a FoodTransaction then the user can sign the transaction with its passphrase. After the transaction is included in the blockchain then is possible for anyone to retrieve transaction information already included in the blockchain. However, only the customer who requested food can use the web application to decrypt sensitive information of its transactions utilizing its passphrase. Also, a restaurant can retrieve customers' food transactions to retrieve sensitive information that was encrypted in specific fields of a FoodTransaction utilizing its restaurant public key. This way only the restaurant and customer can have access to customer-sensitive information even if a blockchain node was exposed publicly allowing anyone to request its API for consulting the transactions in the blockchain (ALVES, 2021b).



Solution architecture (representation)

ISAIC

Users

Web application Restaurant 1

Restaurant 1 backend

Regular node

Seed peer 1

Seed peer 2

Web application Restaurant 2

Restaurant 2 backend

Private route communication between sidechain nodes

Figure 4.2 First restaurant solution architecture (ALVES, 2021b)

4.4.3 Backend Application

(ALVES, 2021b) informed that a backend of a restaurant contains public methods exposed through an API developed with NodeJs technology. The backend application communicates to a regular node of the blockchain, and also it is exposed to allow the restaurant website or restaurant web applications to perform requests to it. The main methods from the back-end API are described below:

- list: List a restaurant food menu option for the restaurant wallet address
- foodDetail: returns details of food by food id
- storeQrCode: generates QrCode that can be used to request food using a mobile phone
- transaction: it searches for a specific transaction ID by transaction ID and customer passphrase
- refund: allows reimbursement of a food request creating a RefundTransaction
- clientPayment: allows payment using transactions signed on the client-side
- payment: allows payment using transaction signed on the backend side
- cryptography: allows to encrypt any message using restaurant backend passphrase and returns an encrypted text, only the same backend can retrieve decrypted data of this encrypted method

4.4.4 Sidechain node on the first solution, also called legacy solution

Sidechain nodes allow communication between them through a private network and subnets, and they reside in different data centers. The seed peer nodes in Figure 1 are the unique nodes that allow the discovery of new nodes in the sidechain, this way each regular node must be connected to a seed peer node to discover other peers in the network and communicate with them. Also, a seed peer node is not exposed to the public network in the solution described by (ALVES, 2021b).

4.4.5 Limitations of the first solution

The first solution of Proof-Of-Concept (PoC) Lisk Restaurant does not have an integration with the Lisk mainnet network and this is its major drawback. Also, on each food request, only one item can be included in a transaction, which means that if the user decides to choose 2 items in the food menu of a restaurant then 2 transactions will need to be created. However, it served as a great example of how would be possible to integrate an industry niche with blockchain technology. Also, it served well as an introductory verification of the potential of blockchain for providing privacy for customers. The version is available on the internet [2].

4.5 IMPROVEMENTS FOR THE NEW RESTAURANT SOLUTION ON SDK 5.X

As described on (ALVES; GREVE, 2021) The improvements included in SDK 5 were concentrated in the introduction of the following main features, consensus mechanism Lisk-BFT, voting mechanism, Lisk Merkle Tree, dynamic fee feature, improvement of messages transmission, block structure, the introduction of modular architecture which can be included or not in the configuration of a sidechain and new tools. Furthermore, it is possible to verify the rationale of each new feature on Github [3]. Regarding the improvements, it was already proven the increase in performance in the Lisk Restaurant solution migrated from SDK 2.3.8 to SDK 5. However, this work shows that is possible to improve even more the performance of Lisk Restaurant with the arrival of a new transaction type that will be described in the next Section. Beyond all the mentioned features, there are still features under development as the interoperability of chains, BLS signature (JUSTIN, 2018), and more. Furthermore, on SDK 5 a sidechain node exposes its data through Interpersonal communication (IPC) or Web Socket (WS) or Hyper Text Transport Protocol (HTTP) connections.

4.6 SECOND SOLUTION

This solution represents the evolution of the first solution described in Section 4.4. The new Lisk restaurant solution sidechain [4] [5], developed and tested with SDK 5.X, allows a

[2] https://www.liskrestaurant.com

[3] https://github.com/LiskHQ/lips/tree/main/proposals

[4] https://github.com/davilinfo/MasterThesis

[5] https://newversion.liskrestaurant.com

customer to request food from restaurants that participate in the restaurant sidechain. For example, if customer A wants to request a meal online from Restaurant B then it must perform a food request on the Restaurant B Web Site. In doing so, the Restaurant Web Site will perform some activities to process the request, for example, create a Food transaction object specifying as the recipient address the public digital address from restaurant B. Beyond that, the restaurant sidechain has some other specific custom transaction types that help to perform such food requests, despite that such a sidechain still allows the regular transaction types that Lisk SDK offers for example the transfer transaction type. One specific characteristic of a custom transaction type is that it allows specifying the fee cost of the transaction, so a custom transaction type can be zero fee or low fee cost depending on the implementation scenario. In the proposed restaurant solution such fee characteristic is trivial and only the sidechain fee is applied and implemented directly in the custom transaction types.

4.6.1 Custom transaction types

Each new custom transaction type must be registered in the sidechain to become available to be utilized by any sidechain node, and therefore by any restaurant connected to the restaurant sidechain. A custom transaction type allows a sender address to send tokens to a recipient address for example, however with custom characteristics defined by the developer of it. Also, the following subsections will present the custom transaction types of Restaurant Sidechain. Below is the basic definition of a custom transaction type structure:

- ModuleID: the module ID where a transaction type is registered. The module then is registered in a sidechain, therefore allowing custom transaction types to be utilized in the sidechain

- AssetID: the asset identification. This is unique and each transaction type should be different from the others

- Nonce: number associated with the sender account of the transaction and it is increased on each new transaction

- Fee: the necessary fee to execute the transaction

- SenderPublicKey: stores the public key of the pseudo-user that signed the transaction with its private key

- Asset: custom data structure specified on each custom transaction type. It is stored and encoded in hexadecimal. In food transaction type it is also encrypted on the customer-sensitive data portion

- Signatures: the digital signatures from the pseudo-user(s) that signed the transaction (LISK, 2021a).

4.6.2 Custom Food Transaction Type

The FoodTransaction transaction type allows a customer to request food online from a specific restaurant. Each restaurant that participates in the sidechain has a Lisk address. Each restaurant respects the privacy of customer-sensitive information encrypting it on each new transaction performed on the client side and broadcasts a signed transaction into the sidechain backend with the customer-sensitive information already encrypted, there is no necessity to share a passphrase. Only the restaurant included as the recipient of a transaction can decrypt a customer's sensitive information.

The main difference from the previous version of FoodTransaction is the design of it (ALVES, 2021b). The new version of Proof-Of-Concept (PoC) Restaurants removed from the FoodTransaction the data fields related to the encrypted information from the customer that only the customer could read and included such information in a new transaction type called ProfileTransaction. The new FoodTransaction type remains with the data fields related to the customer-encrypted information that only the restaurant can decrypt. Also, it included an array of foods that included all the food selections from a customer. Therefore, if a user decides on multiple items in a food menu then only 1 transaction will be placed with all the chosen items. Following is the design of the FoodTransaction asset property:

- items: an array of foods
- price: the total price of foods selected by a customer and included in the items property of the transaction
- restaurantData: encrypted customer sensitive information
- restaurantNonce: nonce utilized to decrypt the customer-sensitive information

The FoodTransaction extends the BaseTransaction type, therefore the regular mandatory fields of a transaction remain. When a food transaction is intended to be signed it is validated before completion upon the method validation. The asset items portion of FoodTransaction is parsed as JSON objects and then validated. If the validation is not successful then a transaction cannot be executed.

Listing A.1 in the appendix has the Javascript helper library method created in this work for creating a FoodTransaction with the encryption of customer-sensitive data. At a glance, the restaurant address is converted to the Base32Address utilizing the cryptography library from Lisk Software Development Kit (SDK). Then, the passphrase provided by the customer is utilized for the retrieval of the customer's digital address and the customer's public key. After it, it is calculated the current account nounce from the customer's digital address. The items collection is iterated for the calculation of the total price of the food request as a request can accept more than 1 item (for example meal and beverage). Then, the customer-sensitive data is encrypted utilizing the customer passphrase, and the restaurant public key with a cryptography library from Lisk SDK. Finally, the custom transaction created in this work is created by providing all necessary information and signing it with the customer passphrase. A send method provided by the created library is utilized to send the transaction into the blockchain network. The customer passphrase was never shared in the network or shared with anyone, all source code that interacted with the passphrase of the customer was at the Javascript level, therefore always on the client side.

The validation method of the FoodTransaction registered in the Restaurant sidechain can be verified upon Listing A.2 in the appendix.

4.6.3 Solution Architecture

For the solution was created the Food, Menu, Profile, and News custom transaction types, and then was created a sidechain with the Lisk tool (Lisk commander [6]). Once the sidechain was created the registration of the custom transaction types was performed, after that, it was created javascript files and a javascript helper library for allowing integration between the external world with the sidechain, and finally, it was created a frontend application integration with the sidechain node. The result can be found in Figure 4.5 and the solution architecture can be first understood in Figure 4.3.

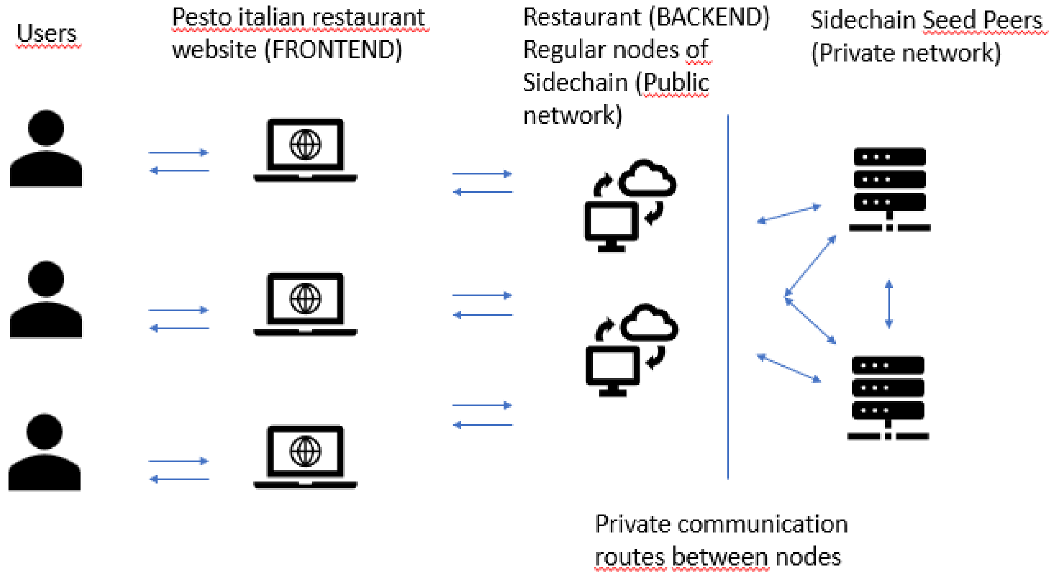

Users
Pesto italian restaurant website (FRONTEND)
Restaurant (BACKEND) Regular nodes of Sidechain (Public network)
Sidechain Seed Peers (Private network)
Private communication routes between nodes

Figure 4.3 It shows the flow of a food request in the second solution.

The components of the restaurant solution are:

- Users: They can perform food requests on a restaurant website
- Frontend: The restaurant website[7] frontend is built with ReactJS[8]. The frontend

[6] http://www.lisk.com for reference

[7] http://newversion.liskrestaurant.com version for demonstration purposes

[8] ReactJS is a javascript library for building user interfaces. https://reactjs.org/

allows users to request food through its interface performing Hyper Text Transport Protocol (HTTP) and Web Socket (WS) requests to regular sidechain nodes. The communication is protected with Secure Socket Layer (SSL) certificate.

- Regular sidechain nodes: These are public sidechain nodes that can run all custom transactions described in this Chapter. They expose HTTP Application Programming Interface (API) and WS API for communication, especially with frontend websites. Also, these nodes communicate with Seed Peers nodes to discover new sidechain nodes that run the blockchain protocols

- Sidechain Seed peer nodes: These nodes allow the discovery of new sidechain nodes connected to them. They can be reached on a private network or public network in this solution.

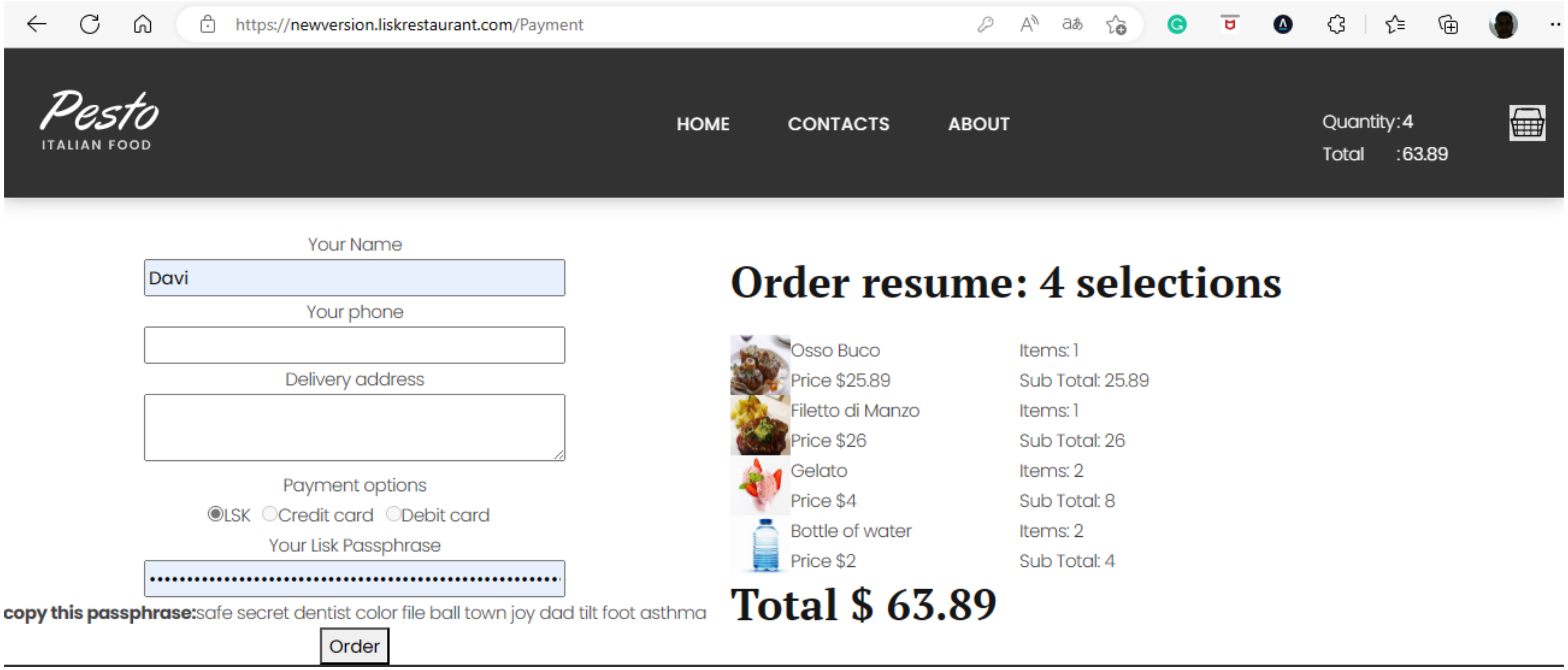
The moment users access the restaurant website they need to choose a meal. Once, the choice is made they request the meal with the following sequence of actions:

1. User chooses a food in the Restaurant Web Site. An important difference between the new version of food transaction in the proposed sidechain to the previous version is the possibility to include several items in a single transaction, Figure 4.4 shows a meal request with several items

2. A user decides for paying the food, then the user provides its Lisk passphrase and requests the food. At this moment, a food transaction is signed with a user passphrase on the client side(javascript), and only the signed food transaction is sent to the sidechain node through a WS request. The user passphrase never reaches a public network, it remains only on the client side. The signed transaction is protected by its hash value, so if any character in the signed transaction is changed then any trustful sidechain node can invalidate the tampered transaction, and not propagate it in the network

3. When a transaction reaches a regular node, the node validates the transaction, places it in its transaction pool, and propagates the transaction in the sidechain network to all connected nodes

4. A delegate, on its round, will propose a block with transactions from the transaction pool including a user food transaction. After that, the consensus mechanism takes place and performs its steps of validation. Lisk-BFT requires two types of messages for a full block validation and finalization in the chain, PREVOTE, and PRECOMMIT messages. Such two types of messages are transmitted between sidechain delegates' accounts. The PREVOTE message can be explained as PREVOTE(B, T, P) where B represents the hash of the block proposed by the delegate and it contains the user transaction, T represents the tip of the chain of the block proposer or the most recent block hash in the chain of the block proposer and P represents the block proposer public key and signature. For any PRECOMMIT message from the block proposer, it is necessary to have a majority of PREVOTE messages $>$



Quantity:4
Total :63.89

Your Name
Davi

Your phone

Delivery address

Payment options
● LSK ○ Credit card ○ Debit card

Your Lisk Passphrase
••

copy this passphrase:safe secret dentist color file ball town joy dad tilt foot asthma

Order

Order resume: 4 selections

Osso Buco	Items: 1
Price $25.89	Sub Total: 25.89
Filetto di Manzo	Items: 1
Price $26	Sub Total: 26
Gelato	Items: 2
Price $4	Sub Total: 8
Bottle of water	Items: 2
Price $2	Sub Total: 4

Total $ 63.89

Figure 4.4 Performing the payment of a meal request.

2/3 from block proposers (HACKFELD, 2019). As explained by (HACKFELD, 2019) and as an example, all > 2/3 honest block proposers can send PREVOTE messages to a block, B1, and these messages reach all block proposers before the next block, B2 considering the block interval. Also, the honest proposer of the next block can include these prevote messages for B1 into block B2. After receiving B2, all honest block proposers can now send a PRECOMMIT(B1) message. Hence, B1 receives precommits by > 2/3 of the block proposers and any block proposer with a decision threshold in (1/3, 2/3) will decide for B1 and all ancestors of B1. Finally, by default, it is only necessary for 10 seconds to include a block in the blockchain or 5 seconds if it was adjusted the time to generate a block in the proposed solution

5. In a scenario without problems at the moment a block is included in the chain with the proposed transaction then it is in the blockchain ledger. From this moment the requested restaurant can start to prepare the requested food and the delivery step;

In the Listing A.3 is possible to visualize several foods included in a single transaction that was executed and included in the block height 86116 of the sidechain [9]. The payload property of such block contains the related transaction id: 27faabfaea7c9c8bafc677519dac13c3dd9216b21bfad724857057a2d1893df2.

4.6.4 Custom Menu Transaction Type

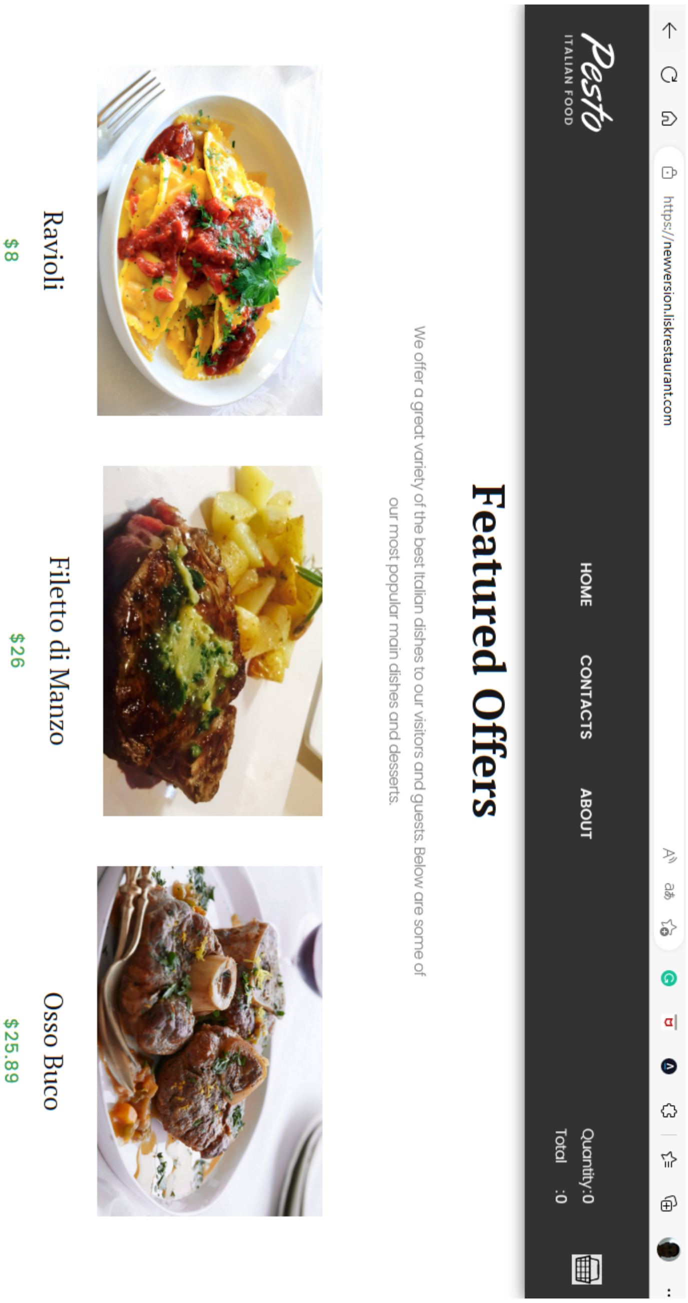
MenuTransaction allows the retrieval of an entire food menu option from a Lisk address belonging to a restaurant just like the first version of Lisk Restaurant (ALVES, 2021b).

The MenuTransaction extends the BaseTransaction type, therefore the regular mandatory fields of a transaction remain there. Also, it is very important to highlight how a Menu transaction is created. Firstly, when a restaurant owner decides to generate a Menu transaction he/she needs to inform the same digital address belonging to the restaurant as the sender and the recipient of the transaction. Such characteristic gives entire control to the restaurant ensuring that only the restaurant can change its menu. Second, when the Menu transaction is signed it proves the ownership of the menu by the utilization of the private key from the restaurant owner. In Figure 4.5 is shown the menu of a restaurant website integrated with the restaurant sidechain. Listing 4.3 has the helper library method for creating a Menu transaction.

Listing 4.3 Helper method for creating a menu transaction by a restaurant

```
async createMenuAssetAndSign(menu, credential){
    const sender = cryptography.getAddressAndPublicKeyFromPassphrase(
credential.passphrase);
    var accountNonce = await this.getAccountNonce(sender.address);
    const tx = await transactions.signTransaction(
        menuSchema,
        {
            moduleID: 2000,
```

[9] http://newversion.liskrestaurant.com:4000/api/blocks?height=86116



Featured Offers

We offer a great variety of the best Italian dishes to our visitors and guests. Below are some of our most popular main dishes and desserts.

Ravioli
$8

Filetto di Manzo
$26

Osso Buco
$25.89

Figure 4.5 In the figure is shown a menu of a restaurant that was loaded from a menu transaction of the restaurant blockchain address.

```
            assetID: 1060,
            nonce: BigInt(accountNonce),
            fee: BigInt(0),
            senderPublicKey: sender.publicKey,
            asset: {
                items: JSON.stringify(menu),
                recipientAddress: sender.address
            },
        },
        Buffer.from(networkIdentifier, "hex"),
        credential.passphrase);
    return tx;
}
```

Furthermore, the validate method of menu transaction can be verified upon Listing A.4.

4.6.5 Custom Encrypted Profile Transaction Type

The ProfileTransaction encrypts customer-sensitive data, this way the customer can retrieve its own transaction data from a single transaction instead of including repeatedly its data on each new FoodTransaction. Furthermore, it saves block space for the inclusion of more transactions. The profile transaction can help a restaurant solution to store and retrieve customer information allowing a customer to save its data in a safer way preserving its sensitive information without having access to a private key of the customer, only a customer can encrypt and decrypt information from a profile transaction.

The ProfileTransaction extends the BaseTransaction type, therefore the regular mandatory fields of a transaction remain. The validate method of ProfileTransaction can be verified upon Listing A.5.

4.6.6 Custom News Transaction Type

NewsTransaction type allows a restaurant to create news and spread it on the blockchain network. Any client application connected to restaurant sidechain nodes would be able to read the news of a restaurant. It is a way to foster restaurant news on a food platform. Despite that, the sidechain owner can also send messages through a restaurant or even customers utilizing the News transaction. Finally, such transaction type allows communication between any participant on the network.

The NewsTransaction extends the BaseTransaction type, therefore the regular mandatory fields of a transaction remain. Listing 4.4 has the helper library method for creating a News transaction.

Listing 4.4 Helper method for creating a news transaction in the sidechain

```
async createNewsAssetAndSignTo(news, credential, lskAddress){
    const sender =
cryptography.getAddressAndPublicKeyFromPassphrase(
credential.passphrase);
    var recipientAddress =
cryptography.getAddressFromBase32Address(lskAddress);

    var accountNonce = await this.getAccountNonce(sender.address);

    const tx = await transactions.signTransaction(
        newsSchema,
        {
            moduleID: 2000,
            assetID: 1080,
            nonce: BigInt(accountNonce),
            fee: BigInt(0),
            senderPublicKey: sender.publicKey,
            asset: {
                items: JSON.stringify(news),
                recipientAddress: recipientAddress
            },
        },
        Buffer.from(networkIdentifier, "hex"),
        credential.passphrase);

    return tx;
}
```

The validate method of NewTransaction registered in the Restaurant sidechain can be verified upon Listing A.6.

4.6.7 Sidechain plugins

Plugins can be attached to a node to provide service information from the node. For example, the HTTPAPIPlugin is a plugin that exposes node information through an Hyper Text Transport Protocol (HTTP) API allowing requests directly to the node that is attached to the plugin. The HTTP API allows searching for specific information on the blockchain such as height, finalized height, registered transactions, delegate accounts, connected peers, and several other node information.

The new solution provided in this chapter exposes an HTTP API on each node of the sidechain through a plugin, therefore it allows for searching node information in the blockchain.

The HTTP API plugin list of available methods requests can be found at (LISK, 2022).

4.7 SUMMARY

This Chapter presented the first solution of the restaurant sidechain and the proposed solution of the restaurant sidechain. The performance evaluation of the proposed solution and the comparison between the proposed restaurant solution and the legacy restaurant solution can be verified in Chapter 5. Furthermore, this chapter presented the encryption method utilized for providing privacy on sensitive data on a custom Food transaction and also it showed a Food transaction JavaScript Object Notation (JSON) created in the website of a restaurant integrated with the proposed solution. Finally, the solution architecture of the proposed solution was described with the information of the items created in this work.

Chapter

5

PERFORMANCE AND SCALABILITY EVALUATION

In this chapter, the sidechain restaurant solution, described in the previous chapter, is experimentally evaluated. Different evaluation scenarios are set up so as to verify how the application behaves in terms of throughput and maximum time user transactions take to be inserted in a block. These metrics are measured for a different number of users, block time, and block payload also called blockchain parameters. During the experiments, several Food transactions were created and sent to the sidechain.

In the second part of this chapter, Section 5.7, the Lightning network, an alternative technology is evaluated in terms of waiting time that micropayments are executed in a channel between two nodes and they got updated in the Bitcoin Testnet network, such a technology can be utilized by merchants and customers. Finally, the structure of the second part of the chapter is different from the first part because the technologies of Lisk and Lightning networks are different, while Lisk has sidechain technology, Lightning is not a sidechain, in fact, it is an off-chain technology dependent on the Bitcoin network.

The chapter ends with a discussion of the technologies evaluated and defines the advantages and disadvantages observed during the experiments.

5.1 CURRENT TECHNIQUES AND EVALUATION METRICS FOR BLOCKCHAIN PERFORMANCE EVALUATION

Before jumping into the Evaluation Aspects section of this chapter we will verify what are the current techniques, and main evaluation metrics for blockchain performance evaluation. As stated by (FAN et al., 2020) there are different ways of measuring performance evaluation in blockchains. The preferred is utilizing real blockchain networks where the full protocol is in utilization by the nodes of the network and real problems can be identified. However, there are cheaper alternatives, for example, the simulation of environments by the utilization of tools that try to mimic the blockchain protocols and consensus mechanism creating an abstraction of protocol rules. Despite, the simulation of the environment is not the most accurate technique for performance evaluation it can be categorized in empirical study analysis based on observation, experiments for example, and also in analytical modeling of performance that utilizes mathematical tools to formalize blockchain system in an abstract way and solve models with rigor.

5.2 SUPPORTING TOOLS FOR DEVELOPMENT AND EVALUATION OF LISK SIDECHAIN SOLUTION

Lisk SDK 5.0.3[1] was utilized in all nodes to implement the sidechain solution. Visual Studio Code[2] was utilized for source code implementation of custom transaction types, libraries, and the sidechain solution. NodeJs was utilized [3] in version 12.22.9 for the proposed solution, it is the engine that allows the execution of sidechain solution, NPM[4] 6.14.15 that allow management of javascript and typescript libraries. A development network was created for the solution. It was captured PCAP[5] files allowing network data to be analyzed with wireshark[6] or reproduced with tcpreplay[7] for better comprehension of the solution developed. The set of delegate accounts utilized on each scenario, the source code, and the experiments can be found on Github[8].

5.3 EVALUATION ASPECTS LISK SIDECHAIN SOLUTION

The goal of this section is to describe our evaluation, its relevance, and its contribution to this work. There are three key aspects that need to be considered in such kind of evaluation, namely throughput, waiting time, and stale blocks. The former two aspects are related to performance whereas the latter refers to keeping consistent information in the system although all three are connected.

As for performance, throughput measures the number of transactions within a block per time (e.g. seconds) whereas waiting time is the time the user waits for its transaction to be included in a block. For most known blockchains, such as Bitcoin and Ethereum, limited throughput is a bottleneck, as already observed in the literature. This also applies to waiting time.

As observed by (AKBARI et al., 2020), intuitively one may try to increase throughput by enlarging block size and shortening the time to build a block, aka block interval or block time. However, doing so may raise the number of stale blocks, which may compromise information consistency. Further, stale blocks, those created outside the main chain, may possibly give rise to chains that are without or are subject to late validation.

The waiting time parameter is important from the customer's perspective as it directly refers to the moment of receiving a food request that a restaurant should start to prepare. Also, the waiting time parameter is important from the restaurant's perspective as it is

[1] https://github.com/LiskHQ/lisk-sdk/tree/v5.0.3

[2] https://code.visualstudio.com/

[3] https://nodejs.org/

[4] https://www.npmjs.com/package/npm/v/6.14.15

[5] https://fileinfo.com/extension/pcap

[6] https://www.wireshark.org/

[7] https://tcpreplay.appneta.com/

[8] http://www.github.com/davilinfo/MasterThesis

related to the moment a transaction was executed to the moment of delivering its result (in our case, meal delivery).

Experimental evaluation carried out in this section seeks to characterize the behavior of the restaurant sidechain application in different scenarios for various values of three key parameters (block interval, block payload size, number of users) observing their effects on throughput, waiting time, and stale blocks. We note that these metrics, already utilized in other scientific papers (AKBARI et al., 2020), were chosen due to their relevance in a restaurant application in which customers request food and need to wait for the payment processing that is represented by a transaction included in a block.

5.3.1 Metrics and parameters

More precisely, in this chapter, we use the following definitions of our evaluation metrics:

- Throughput: Measured as the number of transactions that can be included per block per second
- Maximum waiting time: Maximum waiting time of a user waiting for its transaction to be included in a block
- Stale blocks: The number of blocks that are not included in the correct chain (longest) due to contradiction or concurrency

The evaluation will be carried out taking into consideration varying the following parameters:

- Users: Defined as the number of accounts
- Block interval or block time: Time to forge a block in seconds
- Block payload size, payload for short: The total size capacity in KB of transactions included in a block

5.4 EXPERIMENTAL SET-UP LISK SIDECHAIN SOLUTION

Recall from Chapter 4 the information in a transaction:

- moduleID: module unique ID. Every transaction type belongs to a module. At a module resides the life cycle definition of a transaction and block
- assetID: transaction unique ID
- nonce: a sequential number representing the transaction order that takes place in an account
- fee: fee to execute the transaction
- senderPublicKey: the public address of the transaction owner

- asset: the properties exclusively from Food transaction type.
  - items: Array
  - price: items price sum
  - restaurantData: encrypted data that only the restaurant can decrypt. It contains the user name, phone number, and delivery address
  - restaurantNonce: utilized to decrypt the information of the encrypted restaurantData field
  - recipientAddress: public address of the restaurant

Hence, the asset property on Food transaction type can vary from one transaction to another in a real scenario. However, the experiments were carried out using a fixed transaction so as to standardize the results. Indeed, the transaction asset size is important to determine the transaction size and therefore the number of transactions that can be included in a block. However, a block can have distinct types of transactions with different sizes. Listing 5.1 shows the fixed-format food transaction asset from a food order request to evaluate the metrics in relation to the parameters. Note that when a transaction is created, its attributes are encrypted. In Listing 5.1, non-encrypted information is given for illustration purposes only.

Listing 5.1 Food transaction asset representation format

```
{ items:[{name: "Black Pasta", foodType: 1,
quantity: 1, price:0.1, observation: ""}],
username: "davi", deliveryAddress: "Salvador, number:aaaa,
phone: "71997035287"};
```

During the experimental evaluation, the same asset properties are utilized in a food transaction and performed by different users, taking into consideration various values, for example, the number of users (from 64 to 200), block time (5 and 10 seconds), payload size (from 15 to 30KB) and the number of nodes (4 and 5). Transactions per user and the number of delegates were kept fixed in all experiments, 1 and 103, respectively. These parameter ranges seem to suit the target sidechain under evaluation since we do not expect figures to be distant from actual operational parameters.

The following locations were defined for configurations with 4 nodes. Table 5.1 gives the configuration with 4 nodes: two nodes in Amsterdam and one node each in Salvador and London. The total number of delegates (103) follows the standard Lisk configurations. These delegates were arbitrarily distributed among the nodes. Further, node n1 was chosen as the seed peer. Note that these configuration decisions do not compromise our experimental evaluation since our goal is not related to fault tolerance but mostly to performance. We also considered a configuration with 5 nodes. In this case, all nodes were defined to be in New York, in the same data center. This latter configuration, in Table 5.2, decreases the number of hops and was considered for better evaluating possible problems when decreasing block time. Note that the number of nodes between 4 and 5 is more related to fault tolerance than to throughput. That is, if fault tolerance is an issue, more sites could be considered.

Table 5.1 Scenarios for configurations with four nodes. Each node was configured with Ubuntu 20, 2GB Memory RAM, 50GB Hard drive, and 1 CPU but the Salvador node runs on Ubuntu 20, 8GB Memory RAM, 226GB Hard drive, 8 CPU

Nodes	n1	n2	n3	n4
Location	Amsterdam	Salvador	Amsterdam	London
Total delegates	39	25	20	19
Seed peer	Yes	No	No	No

Table 5.2 Scenario 7 - Nodes were configured with Ubuntu 20, 2GB Memory RAM, 50GB Hard drive, 1 CPU

Nodes	n1	n2	n3	n4	n5
Location	New York	New York	New York	New York	New York
Total delegates	21	25	20	19	18
Block interval	5s	5s	5s	5s	5s
Seed peer	Yes	No	No	No	No

A set of scripts were defined to carry out the experiments. They provide helper methods to connect with a sidechain node and to interact with the sidechain. Through these scripts, it is possible to: generate a new user account on the sidechain; retrieve information from the sidechain node as any account information; create a Food transaction or any other custom transaction registered in the sidechain; and send a transaction into the restaurant sidechain. Furthermore, the scripts can be configured to prepare the creation of a specific number of user accounts and then each created account creates and sends a transaction into the restaurant sidechain. The produced log files from experiments can also be downloaded from GitHub [9].

5.5 EXPERIMENTAL RESULTS LISK SIDECHAIN SOLUTION

In this Section, we present the results obtained from our evaluation in Subsections 5.5.1, 5.5.2, 5.5.3 and in Subsection 5.5.4 we present the results from the legacy version of Lisk restaurant Proof-Of-Concept (PoC) cited by (ALVES, 2021b). First, we investigate the influence of the number of users and block size on throughput and waiting time. Such characterization is important. Indeed, if a large number of users decide to execute transactions simultaneously in the system, then the system should be able to validate and include such transactions in a block. If the transactions could not all be handled

[9] https://github.com/davilinfo/MasterThesis

in a block, then the remaining transactions should be validated and accommodated in the following block increasing the waiting time for executing a transaction from a user. Additionally, blocking time affects both throughput and waiting time as the blocking time is related to the moment a delegate account should forge a block. Intuitively, the shorter the blocking time, the less time a user waits for a transaction to be included in a block. However, the blocking time value should be set so that there is enough time for a delegate to validate the transactions inside the transaction pool, including them in a block. The number of transactions that can be included in a block is a combination of measuring the blocking time, transaction validation time, transaction asset size, and block payload size.

5.5.1 Evaluation – block interval of 10 seconds

Table 5.3 shows the results when the block interval was set to 10 seconds. As expected, waiting time decreases with a larger block size but increases when the number of users increases. As can be observed, there is a lower bound on the waiting time, which corresponds to the minimum number of blocks necessary for processing the transactions submitted by the users. Further, the larger the block size, the higher the throughput tends to be. This is because larger blocks can include more transactions and require fewer blocks to be processed. As can be observed, the number of users did not cause an impact on throughput. This is because the number of generated transactions was not sufficient to overload the transaction pool.

The Graphics below demonstrate the evolution of transactions in relation to the increase of block payload for a block time of 10 seconds and then the evolution of waiting time in relation to users.

Throughput of Food transactions X Payload (kb)

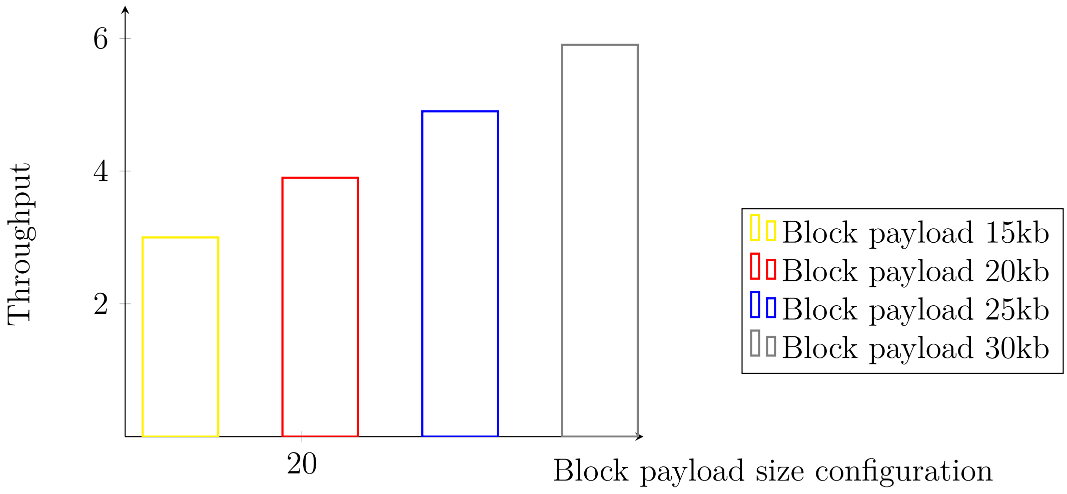
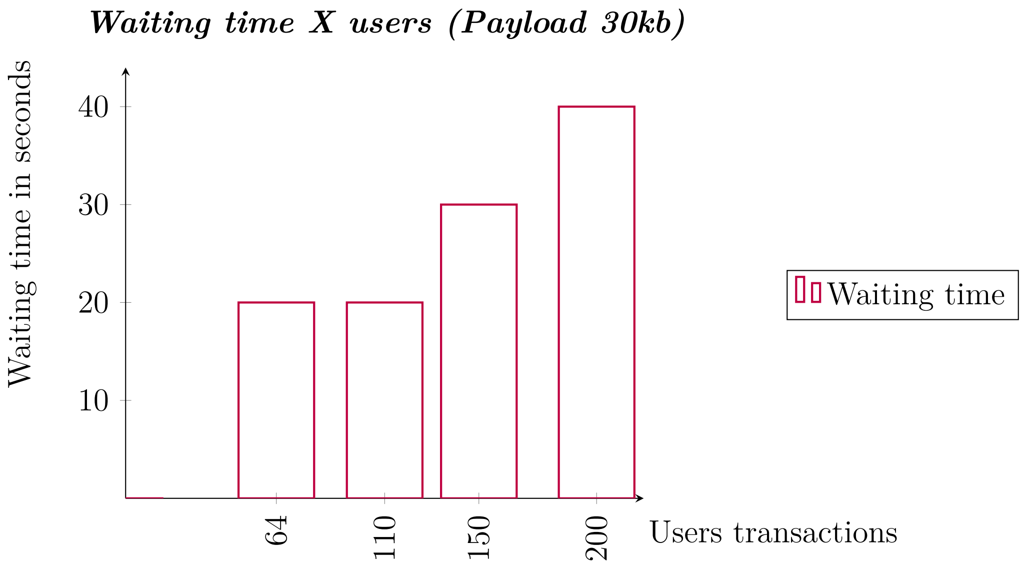


Table 5.3 The following node configuration was utilized to evaluate the performance of the sidechain. Also, each node was configured with Ubuntu 20, 2GB Memory RAM, 50GB Hard drive, 1 CPU

number of users	block payload	waiting time	throughput
64	15	30	3,0
	20	20	3,9
	25	20	4,9
	30	20	5,9
110	15	40	3,0
	20	30	3,9
	25	20	4,9
	30	20	5,9
150	15	50	3,0
	20	50	3,9
	25	40	4,9
	30	30	5,9
200	15	70	3,0
	20	60	3,9
	25	60	4,9
	30	40	5,9



5.5.2 Evaluation – block interval of 5 seconds

Table 5.4 shows the results when the block interval was set to 5 seconds. Similar to the previous results, the waiting time increases with a smaller block size. However, the waiting time is reduced when compared with the block interval of 10 seconds. Also, there is a lower bound on the waiting time, which corresponds to the minimum number of blocks necessary for processing the transactions submitted by the users. The transaction pool was not overloaded during the tests. However, it was observed the creation of stale blocks in these configurations. In fact, during the experiments, nodes struggled to synchronize. It was observed that one node couldn't follow the others and started a single chain. After some time it attempted to sync but then switched back to its own chain and it kept that way. Hence, when a delegate account tried to forge a block on such a node, it created a block in a different chain instead of the correct chain. Interestingly, other delegate accounts configured in the rest of the nodes continued creating blocks in the correct chain, therefore this shows the capacity of the sidechain application to keep working even in the presence of stale blocks as the figures in the evidence table.

5.5.3 Evaluation on a single site – block interval of 5 seconds

The experiments reported in this section focused on evaluating the presence of stale blocks as it is configured with a single site, minimizing the effects due to network communication where five nodes in the same site were considered. It was observed that even under this configuration, setting the block interval to 5 seconds did not remove the occurrence of stale blocks.

Two groups of experiments were carried out. The first followed the configurations presented in Table 5.5. For those configurations, after all transactions were processed, the experiments finished. In this case, no stale blocks were observed. Further, as expected, it was observed that nodes tended to sync faster in a single site when compared to

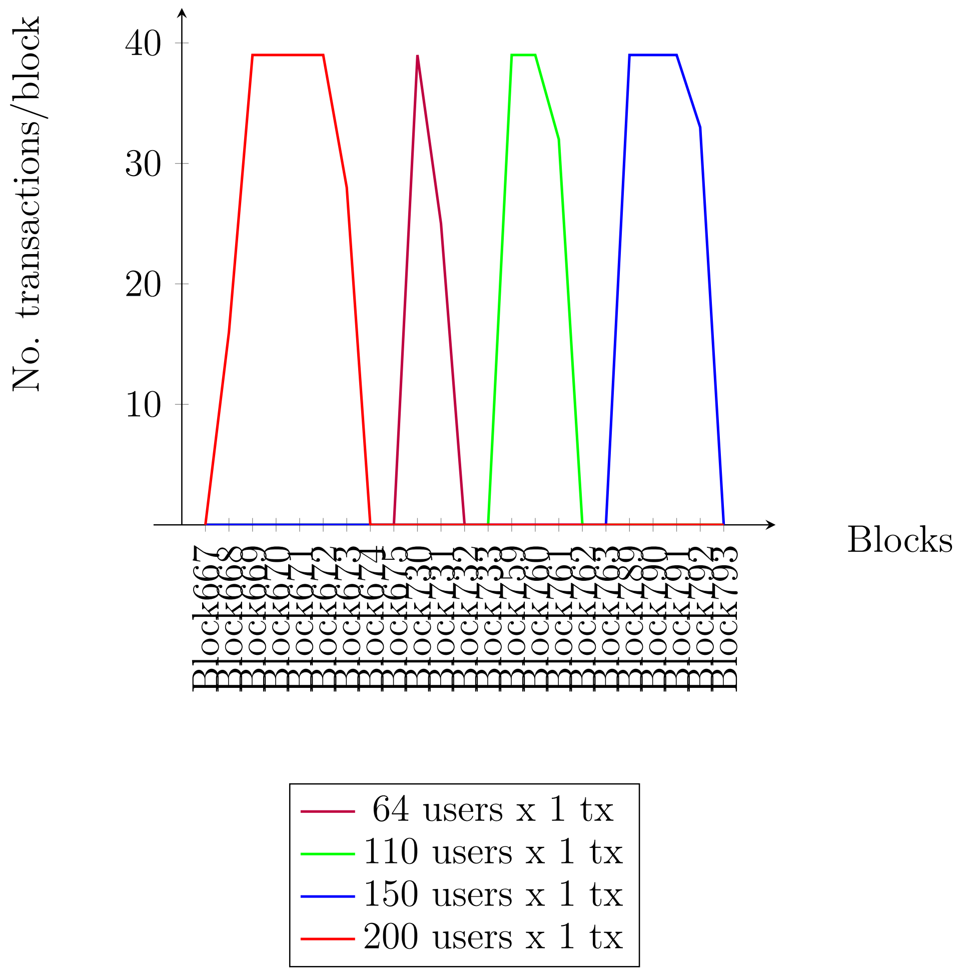
Table 5.4 The following node configuration was utilized to evaluate the performance of the sidechain. Also, each node was configured with Ubuntu 20, 2GB Memory RAM, 50GB Hard drive, 1 CPU

number of users	block payload	waiting time	throughput
64	15	15	6,0
	20	10	7,8
	25	10	9,8
	30	10	11,8
110	15	20	6,0
	20	15	7,8
	25	20	9,8
	30	10	11,8
150	15	25	6,0
	20	20	7,8
	25	20	9,8
	30	15	11,8
200	15	35	6,0
	20	30	7,8
	25	25	9,8
	30	20	11,8

configurations with multiple different geographic locations.

Graph 5.5.3 illustrates the number of blocks necessary to process the transactions for each configuration in Table 5.5. For example, for the configuration with 200 users, a total of 6 blocks were processed. Note that this is exactly what would be expected since the maximum number of transactions per block (39) makes the necessary number of blocks $200/39 = 5.12$.

Scenario 7 - Maximum (tx/block) x users



The second group of experiments consisted of carrying out the previous ones during a longer period of time. Instead of finishing the experiment after all blocks were processed, in this group, the experiments finished after observing the block of number 230, 000. In this case, it was observed the creation of stale blocks. For example, in one of the configurations (n1 configured as seed peer and forger with 2GB RAM, 1 CPU, Ubuntu 20, 50GB Hard drive, 21 delegate accounts, the backup of n1 and n2 can be found on GitHub [10]), node n1 was capable of finalizing a block until the height 3064 but no other block after that as it became disconnected from other nodes. It then initiated the creation of its own chain, creating more than 100, 000 blocks with only 21 delegate accounts. Those

[10] https://github.com/davilinfo/MasterThesis

blocks would never become finalized in the network as the number of required delegates would be missing. Nodes n2, n3, n4, and n5 continued as participants in the main chain. They were able to create and finalize more than 200, 000 blocks but slower than what would be expected due to the fact that 21 delegate accounts were configured at node N1. Clearly, these 21 delegate accounts were not able to forge any block in the correct chain. Note that the presence of stale blocks may cause a decrease in throughput due to this kind of effect.

Table 5.5 Scenario 7 - Evaluation of the maximum number of Food Asset transactions included in a block and maximum waiting time by a node that received the transactions. Each node was tested by time utilizing a spam script. 5s block interval, 20kb block size. Differently from Scenario 5, this experiment utilized 5 nodes, adding fault tolerance to the sidechain

Users	maximum nº of Tx/block	Waiting time
64	39	10 seconds
110	39	15 seconds
150	39	20 seconds
200	39	30 seconds

5.5.4 Evaluation of legacy version of (PoC) Lisk restaurant - block interval 10 seconds

The evaluation was performed with the version of Lisk Restaurant cited by (ALVES, 2021b), and it can include up to 25 Food transactions in a single block, Table 5.6. It is possible to consult each block changing the height of the block with the following link [11].

Table 5.6 Scenario 10 - PoC solution 1 cited by (ALVES, 2021b). Evaluation of the maximum number of Food transactions inside a block. Following, the test was performed with 99 Food requests utilizing a script on a running node configured with a 10s block interval, and a default block size of 15kb. As a result, it was observed the total waiting time of the 60s to execute all 99 Food transactions. The node configuration was similar to other scenarios shown in this work as running Ubuntu 20, 2GB Memory RAM, 18 GB Hard Drive, 1CPU

Height	102152	102153	102154	102155	102156	102157
nº of Tx/block	1	23	25	25	24	1

After testing the legacy version of Lisk restaurant it was possible to compare the rate of a maximum number of transactions per block with the new version in different configurations as shown in Table 5.7.

[11] http://www.liskrestaurant.com:4000/api/blocks?height=102155

Table 5.7 Comparison between PoC solution 1 cited by (ALVES, 2021b) and the second version proposed in this work. It was observed improvements in all scenarios as the Legacy version does not have the option to change the block payload size. Also, the first solution utilized the DPoS consensus mechanism with a simple majority rule of $> 50\%$ validators verification to include a block and no finalization rule. The first version had issues when configured block interval of 5s.

Legacy version max nº of Tx/block	New version max nº of Tx/block	Payload	Tx growth rate
25	30	15Kb	20%
25	39	20Kb	56%
25	49	25Kb	96%
25	59	30Kb	136%

Finally, it was observed that the expected results of performance with the new implementation of the Lisk restaurant sidechain surpassed the legacy version of Lisk Restaurant PoC (ALVES, 2021b) in number of transactions per block by twice at least and also in faster block time creation.

5.6 IMPORTANT OBSERVATIONS OF LISK SIDECHAIN SOLUTION EXPERIMENTS

In this chapter, several experiments were carried out so as to characterize the designed food transaction sidechain application. Some observations are worth highlighting. First, as the experimental results showed, the configurations with block intervals of 10 seconds were stable in terms of the absence of stale blocks. When the block interval was configured to 5 seconds, stale blocks were observed even in single-site configurations.

Second, it was possible to observe for the target application, that setting the block payload to 30 KB is cost-effective. Further, a reasonable number of transactions per block is obtained. For the fixed transaction used in the experiments, 30 KB leads to 59 (30 KB divided by transaction size) Food transactions per block.

The stale blocks effect when the block interval was configured to 5 seconds had the consequences described in the Consistency, Availability, Partition Tolerance (CAP) Theorem in Chapter 2. In the Lisk experiment was proved that a Lisk node would continue retrieving information when requested even when itself was not synchronized with the majority of other nodes and delegate accounts in the chain, therefore, configuring that Lisk is a blockchain that favors availability over consistency. In such a scenario, the trilemma didn't favor the security in the chain as stale blocks were created by small groups of delegate accounts configured in such a node. As described by (AKBARI et al., 2020) the presence of a high number of stale blocks increases the possibility of the appearance of fraudulent activities. For example, if a merchant website is connected to a node such as the mentioned scenario of stale blocks then the transactions received by the merchant in such a node could have no real value as the blocks containing the transactions will never be included and finalized in the longest chain. A simple solution for the merchant is to start the preparation of any meal only when the chain is properly synchronized and the block containing the transaction of the meal request is finalized.

Although some corner scenarios were found, our evaluation indicated that this application can be successfully set up to be of practical use as the observed metrics indicated reasonable operational figures in terms of throughput and waiting time.

5.7 METHODOLOGY OF EXPERIMENTS WITH LIGHTNING NETWORK

The previous part of this Chapter was dedicated to Lisk sidechain evaluation and experiments. From this section until the end of this Chapter the experiments will evaluate the Lightning network and its relationship with Bitcoin blockchain.

The methodology of experiments was organized as shown in Figure 5.1, source (MELO, 2021).

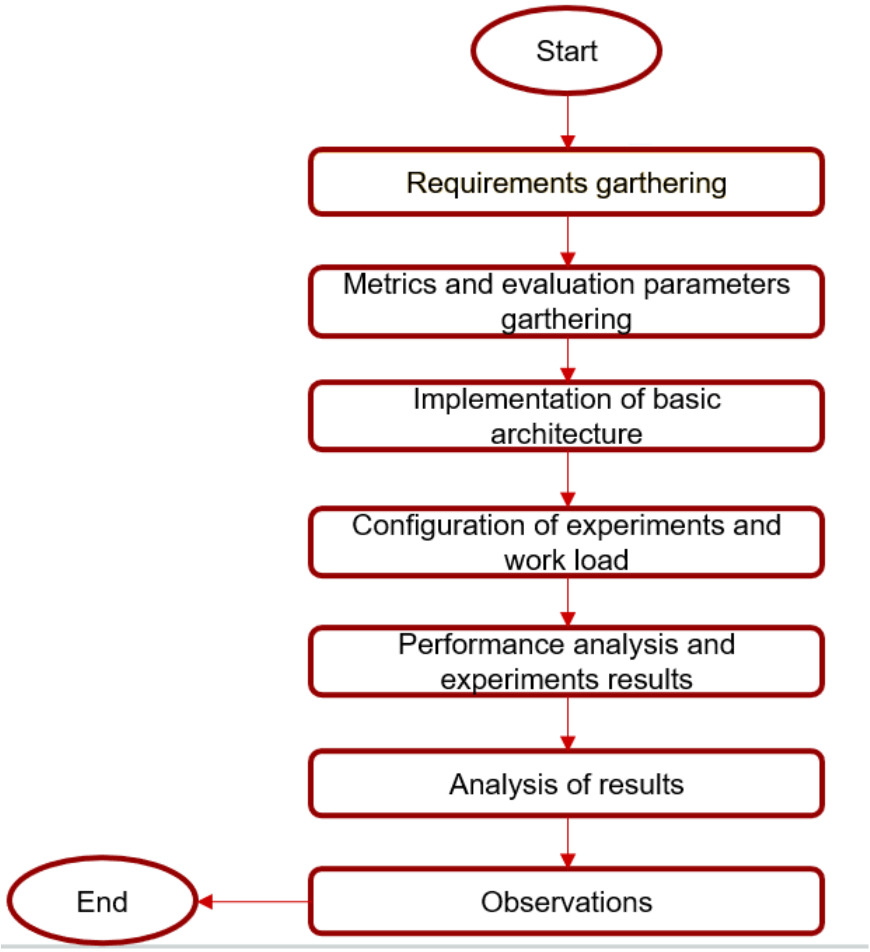

Start
Requirements garthering
Metrics and evaluation parameters garthering
Implementation of basic architecture
Configuration of experiments and work load
Performance analysis and experiments results
Analysis of results
Observations
End

Figure 5.1 Methodology of experiments with Lightning Network. Source: (MELO, 2021)

5.7.1 Requirements garthering

This item defines everything that is required to understand the components of the system, the requirements, and its functionality. It was defined as requirements gathering:

- Pre-requirements: Knowledge of blockchain
- Inputs: Websites, GitHub source code, the blockchain community, Matrix of related works
- Actions: Identification of principal system components
- Products: Requirement list of deployments of the system and its evaluation for fast payments:
    - Download of entire Bitcoin blockchain network into a Lightning node
    - Connection between peers of a channel or intermediaries' peers
    - Creation of channels between 2 peers for processing invoices (Funding transaction – Bitcoin network)
    - Creation of invoices for allowing payments (Mutual agreement)
    - Execution of micropayments inside a channel (Lightning transactions, off-chain)
    - Closure of a channel (Settle the last state of micropayments in a transaction sending it to the Bitcoin network) – it respects the Bitcoin confirmation protocol
    - Withdrawal from Lightning to Bitcoin address – Bitcoin transaction
- Pos-requirements: Comprehension of the system and its functionality.
    - Lightning runs on top of Bitcoin
    - Lightning is not a sidechain (no exclusive blocks)
    - Channels can have one or two directions, require two peers
    - Channels are determined by Hash timedlock contracts
    - Channels can execute several micropayments (routing payment can be required)
    - Withdraw from lightning wallet can be performed directly to Bitcoin address without explicit channels

5.7.2 Metrics and evaluation parameters garthering

This phase identified the principal metrics and parameters evaluation in the blockchain technology based on books and scientific papers and generated as a product a list of metrics and parameters for performance evaluation related to blockchains.

- Pre-requirements: Comprehension of the system and its functionality
- Inputs: Books and scientific papers (POON; DRYJA, 2016; LIGHTNING, 2023a; YANG et al., 2020)
- Actions: Identification of metrics and parameters evaluation
- Products: List of metrics and parameters for performance evaluation related to blockchains
  - Metrics: waiting time: the time how long a user waits for a transaction
  - Parameters: number of micropayments in the receiver channel
- Pos-requirements: Comprehension of metrics for performance evaluation

5.7.3 Implementation of basic architecture

This phase identified the set of hardware and software required for the execution of the system and its evaluation.

- Pre-requirements: Comprehension of the system and its functionality
- Inputs: Set of hardware and software required for the execution of the system and its evaluation. Configured Hyper-V on Windows 11
  - 1 VM 1 GB Ram, 120 GB Hard Drive, 1 CPU, Ubuntu 20.04, Lightning-C, Lightning-cli version 0.12.1, testnet, Bitcoin-cli version 24900, testnet
  - 1 VM 1,2 GB Ram, 120 GB Hard Drive, 1 CPU, Ubuntu 20.04, Lightning-C, Lightning-cli version 0.12.1, testnet, Bitcoin-cli version 24900, testnet
  - Configuration of pm2 program for monitoring Bitcoin and Lightning execution
- Actions: Deployment of 2 Virtual machine (VM)s on Bitcoin Testnet and Lightning Testnet
- Products: Basic architecture. VMs up and running
- Pos-requirements: VMs Bitcoin Testnet and Lightning Testnet in execution

5.7.4 Configuration of experiments and workload

This phase shows how the experiments and workload are utilized for performing the evaluation of the Lightning network with the Bitcoin network.

- Pre-requirements: Virtual machine (VM)s Bitcoin Testnet and Lightning Testnet in execution
- Inputs: Bash script to be executed in a Lightning node

- Actions: Definition of working load to be executed in Lightning
    - creation of 40 invoices representing the merchant by Node 1 and respective scripts for invoice payment utilizing bolt11 by Node 2. A common node will be utilized for rotation in the channel (Lightning faucet node, public key: 0270685ca81a8e4d4d01beec5781f4cc924684072ae52c507f8ebe9daf0caaab7b)
    - Another scenario is funding a node address and then performing a withdrawal from the customer address to the Merchant address
- Products: Scripts for working load injection and log monitoring in lightning and bitcoin testnet
- Pos-requirements: Integrated environment

The first action consisted of writing down the public key of the principal actors (Node 1, Node 2, Faucet node as a rotating node):

- Node 1: "032632259765dd04833258446729d6ef7835c0a7b2dcdf3dcc407d09a477a05ab1"
- Node 2: "0384179270aee78fc4271206f2a8440a9cb81ed06a6172e99b21d79c0b4f4ee3d0"
- Faucet: "0270685ca81a8e4d4d01beec5781f4cc924684072ae52c507f8ebe9daf0caaab7b"

The following sequence of steps shows the configuration of communication between Node 1 and Node 2 of the first action:

0 Synchronization of Bitcoin and Lightning networks on Node 1 and Node 2

1 Creation of bech32 address in Node 1 and Node 2

2 Connecting Node 1 to the Lightning community Faucet[12]

3 Establishing a channel of 100000 satoshis and 0 satoshis as the initial balance from the Faucet to Node 1 - funding transaction sent to Bitcoin network

4 Connecting Node 2 to the Lightning community Faucet

5 Funding Node 2 address by the establishment of a channel from the Lightning community Faucet to Node 2 - funding transaction sent to Bitcoin network

6 Closure of Node 2 channel

7 Creation of Node 2 channel to Faucet node, channel amount of 50000 satoshis

8 Creation of invoices from Node 1, invoices.sh file that generated the file results_-invoices.txt

[12] https://faucet.lightning.community/

9 Payment of invoices by Node 2, pay_invoices.sh file that generated the file result_-payments.txt, Listing A.9

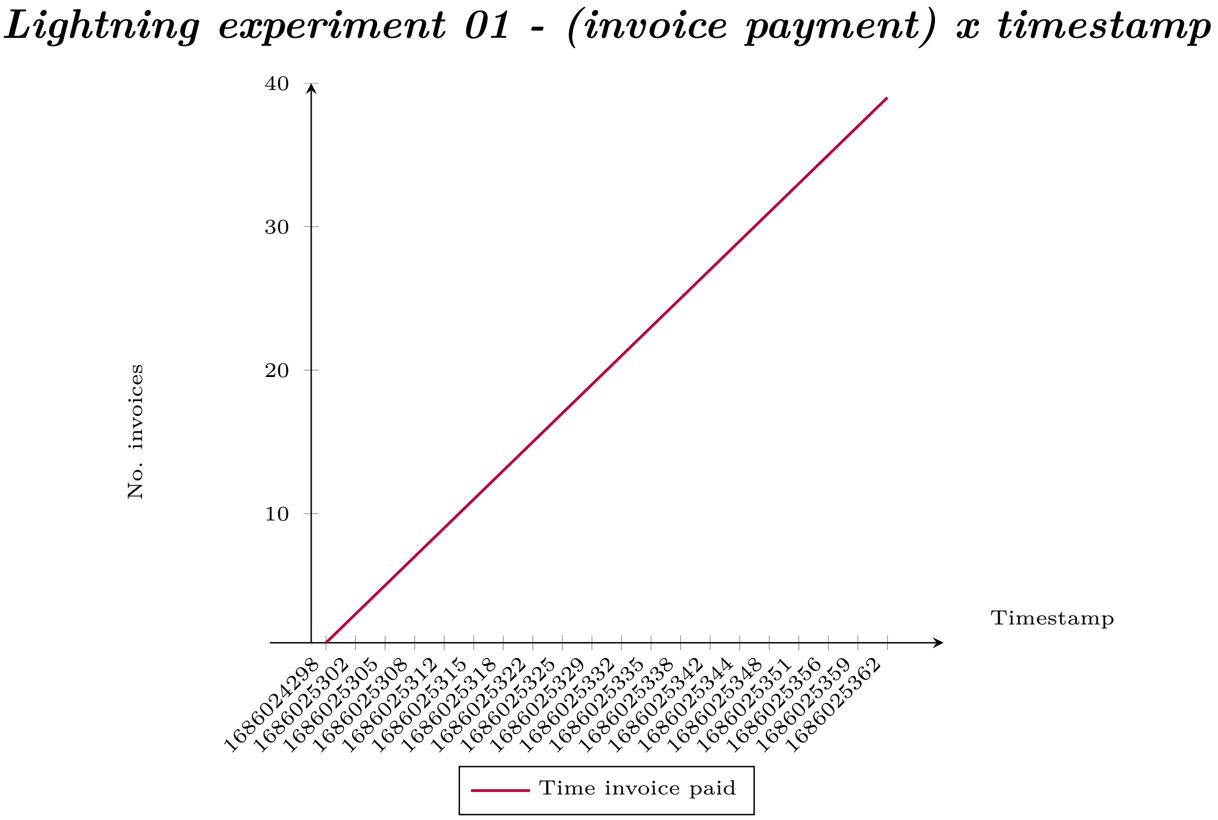
10 Closure of the channel by Node 1

11 Waiting for the channel close transaction to be included on the Bitcoin network, Listing 5.2

During the experiment, it was observed the metric waiting time in relation to the execution of micropayments for all invoices. Considering that it is necessary to inform the utilization of bash scripts to perform the experiments and each script executes synchronously which means that when a first command is executed then the second command of the script will initiate execution only when the first command receives a response, therefore the difference of time between each payment executed by Node 2 was something around 3.3 seconds. The total amount of time to execute all 40 payments can be verified in Table 5.8. The Chart 5.7.4 shows the growth in time from the first payment of the invoice until the payment of the last invoice in relation to time in the timestamp.

Table 5.8 Experiment 01 - Total of waiting time to pay all 40 invoices raised from Node 1 and paid by Node 2

No Invoices	Initial Time Payment 1st invoice	Finish Time Payment invoice 40	Waiting time Lightning	Waiting time Lightning to Bitcoin (Testnet)
40	01:21:38	01:23:50	2m:12s	<=20min



Lightning experiment 01 - (invoice payment) x timestamp

Following are the results of the payment of the invoices in Node 1.

Listing 5.2 Node 1 funds before closing channel. Each invoice value is 2000 * 40 invoices

```
{
  "outputs": [
     {
        "txid":
"5172ff4b529b5aacd372f753f6bc6e26559231718afd6ecde815ae676820c88b",
        "output": 0,
        "value": 5405,
        "amount_msat": "5405000msat",
        "scriptpubkey": "00140afbb37a4bfc38a429add251719eb28747aabf28",
        "address": "tb1qptamx7jtlsu2g2dd6fghr84jsar640egq57r9z",
        "status": "confirmed",
        "blockheight": 2435750,
        "reserved": false
     },
     {
        "txid":
"bbbe939848977a58f75409e31d6c3cda965ae4729e2a5e2c604ae6414f533816",
        "output": 0,
        "value": 2040,
            "amount_msat": "2040000msat",
            "scriptpubkey": "001443dfd348895cea1bb50da877d540d42107
            8b7ce3",
            "address": "tb1qg00axjyftn4phdgd4pma2sx5yyrckl8rk4xezn",
            "status": "confirmed",
            "blockheight": 2436348,
            "reserved": false
        }
    ],
    "channels": [
        {
            "peer_id": "0270685ca81a8e4d4d01beec5781f4cc924684072ae
            52c507f8ebe9daf0caaab7b",
            "connected": true,
            "state": "CHANNELD_NORMAL",
            "short_channel_id": "2436592x55x1",
            "channel_sat": 80,
            "our_amount_msat": "80000msat",
            "channel_total_sat": 100000,
            "amount_msat": "100000000msat",
            "funding_txid": "9060fdc271c8224f7dd93f13ea77fb88f6d66da
            303ab6f0a3d3c80aaf5d7ef72",
            "funding_output": 1
        }
    ]
}
```

Closing the channel generated the Bitcoin transaction bd138d32da0e6584cd098d2f725232de7df2cc3b08a15d371991b5f9e92e7a0b. However, an unexpected behavior happened in this experiment, after the channel between Node 1 and Node 2 got closed and achieved the status of On-chain no output was created with the balance of the channel. Beyond that, some issues happened during the overture of Node 2's channel, for example, freezing status in the channel, in such situation it was necessary to force a closing of the channel and then perform an opening of a new channel, therefore, it consumed several minutes for performing the experiment. Upon Bitcoin Testnet network a block is mined every 20 minutes, and a channel is activated or deactivated only after 1 confirmation in the network.

Knowing the issues of the first experiment, another experiment was carried out utilizing the same basic architecture as the first. A new channel was opened, a new list of invoices was created and payments were executed. The results can be verified in the Listings A.10, A.11, A.12, A.13. Also, the waiting time metric was measured and the results are available in Table 5.9.

The output or UTXO containing the total amount from the invoices of Node 1 that were paid by Node 2 and after closing the Node 1 channel after the experiment can

Table 5.9 Experiment 02 - Total of waiting time to pay all 40 invoices raised from Node 1 and paid by Node 2

No Invoices	Initial Time Payment 1st invoice	Finish Time Payment invoice 40	Waiting time Lightning	Waiting time Lightning to Bitcoin (Testnet)
40	03:58:03	04:00:10	2m:07s	<=20min

be found in the Listing 5.3 of the Lightning Node 1 and also on the Bitcoin address tb1qtjsnqk3v75la59hwqynyvaf94p0je7pfh69mzq.

Listing 5.3 Lightnind Node 1 UTXO after closing channel with paid invoices by Node 2

```
 {
      "txid":
"57f5d1bfc0153f8262755e68d6a032b04b5a6da6defd9b5118d860f288b49dc7",
      "output": 0,
      "value": 3780,
      "amount_msat": "3780000msat",
      "scriptpubkey": "00145ca1305a2cf53fda16ee0126467525a85f2cf829",
      "address": "tb1qtjsnqk3v75la59hwqynyvaf94p0je7pfh69mzq",
      "status": "confirmed",
      "blockheight": 2436734,
      "reserved": false
  }
```

The second action consisted of performing a withdrawal from the customer's address to the merchant's address without the necessity of opening a channel of communication. Listing 5.4 and Listing 5.5 show the same transaction id from Node 2 withdrawal and Node 1 Unspent Transaction Output (UTXO) 6009ba74db368a923ca8fbb85790a309f3e905176b3b1a 4567e83972aaa3474f [13].

Listing 5.4 Node 2 withdrawal to Node 1 address

```
{
    "tx": "0200000001f9306ab4671e24c4748c19a102c8d7e8f525f9638c1a59347
    b99ca3e1051523501000000000fdffffff02f3ba000000000000160014e7c
    2879060892078362903ca8bef815bf4cc3d82d0070000000000001600140
    afbb37a4bfc38a429add251719eb28747aabf28072e2500",
    "txid":
"6009ba74db368a923ca8fbb85790a309f3e905176b3b1a4567e83972aaa3474f",
    "psbt": "cHNidP8BAHECAAAAAfkwarRnHiTEdIwZoQLI1+j1JfljjBpZNHuZyj4QUVI
```

[13] https://blockstream.info/testnet/tx/6009ba74db368a923ca8fbb85790a309f3e905176b3b1a4567e83972aaa3474f

```
    1AQAAAAD9////AvO6AAAAAAAAAFgAU58KHkGCJIHg2KQPKi++BW/TMPYLQBwA
    AAAAAABYAFAr7s3pL/DikKa3SUXGesodHqr8oBy4lAAABAP1aAQIAAAAAAAQE
    TCItxEzWfNVBUq7VIRk/1frVDjXO7tSzixJ2rqqV+EgAAAAAA/////wKawgA
    AAAAAACJRIGCDvQvIbIQcj0yQF0UjhnbGNIwH1huEsjm3E9ycCNgqUMMAAAA
    AAAAWABSOmXuAVC/aEV93s2h2+05LjrK/ZwQASDBFAiEA7OkUzAzPSPjcnZh
    uLMuQb/x1V1GBewZt9TObJGA5cHcCIDvWjx66vFXUaUCzB+89VbW8XotwhRK
    xr84xe9P8XFjXAUcwRAIgfsyCfatYaFpg5r8eJHBlUDuxo4g+lYAiVq3IIKB
    UH/ACIHBmlhyVnUUiKsnJ8X/bhBdOvWgwgLK3doF6QGp5NO8RAUdSIQNxH8q
    YU3cKajR666DT9+GDJuvlgCqd6WgYqJByjdioAyEDe55XuwWlYlD4yx97Prb
    N9UKFXXi7MxpTjyFHmRNY7btSrgAAAAABAR9QwwAAAAAAABYAFI6Ze4BUL9o
    RX3ezaHb7TkuOsr9nIgIDFnh7LXGoHLr+NYJN/3q+u2lZKnBY8TFGAor+Rst
    FTp5HMEQCIHGI5davYQEjV712B0tb0fXSYaP45h9SiRo9S5iI0AxXAiABBcP
    dp1lf9qCdDaOg3Ndu/nzVTRSarfsBXCfC7TTcagEiBgMWeHstcagcuv41gk3
    /er67aVkqcFjxMUYCiv5Gy0VOngiOmXuAAAAAAAAiAgPBiKE/E8OlpMWj8YM
    4MYUmLhxZwbxjGlg5Yq93i+SNpAjnwoeQMwAAAAAM/AlsaWdodG5pbmcEAgA
    BAA=="
}
```

Listing 5.5 Node 1 UTXO from Node 2 withrawal. Compare the txid with the Listing 5.4

```
{
    "txid":"6009ba74db368a923ca8fbb85790a309f3e905176b3b1a4567e
    83972aaa3474f",
    "output": 1,
    "value": 2000,
    "amount_msat": "2000000msat",
    "scriptpubkey": "00140afbb37a4bfc38a429add251719eb28747aabf
    28",
    "address": "tb1qptamx7jtlsu2g2dd6fghr84jsar640egq57r9z",
    "status": "confirmed",
    "blockheight": 2436616,
    "reserved": false
}
```

Finally, a third experiment was conducted to verify the execution of payments of invoices by Node 2 but on different terminals. The goal was to measure the waiting time for executing the micropayments in a concurrent environment. For that, it was created 2 invoices on Node 1, and these two invoices got paid by Node 2 but each was on a different terminal. The execution of payments was initiated at almost the same time as it was necessary for pressing the enter button on the keyboard. The result can be visualized on Listings 5.6, 5.7. Table 5.10 shows the difference in the amount of time between the payment of the first invoice and the payment of the second invoice, just milliseconds.

Listing 5.6 Experiment 03 Node 1 invoices

```
{
 "label": "invoice01_1",
 "bolt11": "lntb150n1pjgyes6sp563px5n3shjwpfewmuvhfuv2kfayahnk860f96
 f9n4kqkyex3mvqqpp5hq2ctv9gvmulj2w6245vxjrdpv2g8ahd4w9rnzgm
 2rm6pfqgymysdquv3jhxcmjd9c8g6t0dcsxyun9v9jqxqrafscqp2rzjqf
 cxsh9gr28y6ngphmk90q05ejfydpq89tjjc5rl36lfmtcv424hkff0ggqq
 qrqqqyqqqqlgqqqqqqgq2q9qyysgqhlanus0dn5zhyym3kgpxuatwvk65m
 he0x63g5dz9s8dl4h5pqfuz66vsnpchnud72jfugnkzk6w8fdp5fyy50wvu
 vpqevjuuhuds8wgp7fn2t8",
 "payment_hash": "b81585b0a866f9f929da5568c3486d0b1483f6edab
 8a39891b50f7a0a40826c9",
 "msatoshi": 15000,
 "amount_msat": "15000msat",
 "status": "paid",
 "pay_index": 91,
 "msatoshi_received": 15000,
 "amount_received_msat": "15000msat",
 "paid_at": 1686267755,
 "payment_preimage": "39a2c98a2cb663218b7ccae53bf6f4157a23645343
 b801ca284dca4f30d0a887",
 "description": "description bread",
 "expires_at": 1686297418
},
{
 "label": "invoice01_2",
 "bolt11": "lntb150500p1pjgyejwsp5rlluzqgc9jl6g653p4ye8xwdxqj5wrwxwgxfv
 75xpzc8h8dky7hspp5sz6uyzjafg57uvasj830sg7p8nmw33xfdzwvtzxnf8y
 m0qkf203sdquv3jhxcmjd9c8g6t0dcsxyun9v9jqxqrafscqp2rzjqfcxsh9
 gr28y6ngphmk90q05ejfydpq89tjjc5rl36lfmtcv424hkff0ggqqqrqqqyq
 qqqlgqqqqqqgq2q9qyysgq0qe2jn485q8xuefnfklawetjccwkttgqfxv867
 6533pmnwckq7rx3kecml3cxc9f290usurtatlgvnvfchur9smpzhun75kakx
 nydggquzkn4r",
 "payment_hash": "80b5c20a5d4a29ee33b091e2f823c13cf6e8c4c9689cc588d34
 9c9b782c953e3",
 "msatoshi": 15050,
 "amount_msat": "15050msat",
 "status": "paid",
 "pay_index": 90,
 "msatoshi_received": 15050,
 "amount_received_msat": "15050msat",
 "paid_at": 1686267755,
 "payment_preimage": "477bb5083f679c8bcdb5e63430c121b862402f5195801e30
 ac874dcb1019270b",
 "description": "description bread",
 "expires_at": 1686297470
}
```

Listing 5.7 Experiment 03 Node 2 paid invoices

```
{
   "destination": "032632259765dd04833258446729d6ef7835c0a7b2dcd
   f3dcc407d09a477a05ab1",
   "payment_hash": "b81585b0a866f9f929da5568c3486d0b1483f6edab8a3
   9891b50f7a0a40826c9",
   "created_at": 1686267751.947,
   "parts": 1,
   "msatoshi": 15000,
   "amount_msat": "15000msat",
   "msatoshi_sent": 16000,
   "amount_sent_msat": "16000msat",
   "payment_preimage": "39a2c98a2cb663218b7ccae53bf6f4157a236453
   43b801ca284dca4f30d0a887",
   "status": "complete"
},
{
   "destination": "032632259765dd04833258446729d6ef7835c0a7b2dcd
   f3dcc407d09a477a05ab1",
   "payment_hash": "80b5c20a5d4a29ee33b091e2f823c13cf6e8c4c9689cc
   588d349c9b782c953e3",
   "created_at": 1686267752.454,
   "parts": 1,
   "msatoshi": 15050,
   "amount_msat": "15050msat",
   "msatoshi_sent": 16050,
   "amount_sent_msat": "16050msat",
   "payment_preimage": "477bb5083f679c8bcdb5e63430c121b862402f519
   5801e30ac874dcb1019270b",
   "status": "complete"
}
```

The Listings A.14, A.15, A.16 shows the closing transaction of the channel on Node 1 at Experiment 03, the on-chain status of Experiment 03 and the output generated after closing the Node 1 channel.

5.7.5 Analysis result in Lightning network

This phase analyzed the results of the experiments performed in the Lightning network. It was configured as one channel between two peers, Node 1 and Node 2, such a channel

Table 5.10 Experiment 03 - Total of waiting time to pay all 2 invoices raised from Node 1 and paid by Node 2

No Invoices	Initial Time Payment 1st invoice	Finish Time Payment invoice 2	Waiting time Lightning	Waiting time Lightning to Bitcoin (Testnet)
2	20:42:31.947	20:42:32.454	0m:00s:507ms	<=20min

was established by mutual agreement. The channel had a total amount of 100,000 while the initial balance was 0. Each invoice was created with the value of 2000, and a total of 40 invoices were created, therefore Node 2 paid 2000 millisatoshis for each invoice generating a balance for Node 1 of 80000, the Listing 5.2 confirms such information. During the experiment was measured the total amount of waiting time for executing the micropayments inside the channel and also to close the channel and send the resultant transaction into the Bitcoin network, such a metric was based on the number of invoices paid and how long it took to perform all payments on Node 2, the result can be verified in Table 5.8. However, it should be explained that the payment script was executed synchronously, which means that the system waited for the response of each invoice payment to proceed to the next invoice payment.

The micropayments experimented with were limited to 40 micropayments, and then 2 micropayments, but it would be possible to execute many more micropayments in a channel, however, for a faster rate result would be necessary the execution in parallel of the micropayments by the utilization of more nodes for a more accurate analysis.

Another experiment was also performed in the Lightning network, it was a withdrawal from Node 2 to Node 1 that generated another Unspent Transaction Output (UTXO) of the value of 2000, Listings 5.4, 5.5. Upon the Lightning Testnet network, and after closing a channel in the Lightning network it is necessary to wait for 1 confirmation to visualize the transaction resultant of the channel on the Bitcoin Testnet network. Beyond that, on Bitcoin Mainnet it is necessary to wait even more time, in such a case for 3 confirmations in the network.

5.7.6 Observations about Lightning network

The lightning network channels can be interesting in scenarios where are parties involved as third-party companies, exchanges, and government for example which can demand several payments in a roll, configuring the necessity of micropayments in a channel. After sending funds to a Lightning node address the withdrawal feature on Lightning becomes a simple and direct transfer method between Bitcoin addresses.

Considering the Lightning experiments it is possible to discuss a scenario in which a merchant can establish shifts of work. For each shift of work, it is necessary to open a channel for receiving payments from customers in the Merchant's Lightning node addresses, and at the end of each shift, it is necessary to close the channel to confirm the payments into the Bitcoin network. A third-party company software could be responsible for providing all the technology of the applications for the merchant and customers, therefore becoming a router between them. Customers would have the possibility for payment by withdrawing directly from their wallet to the merchant's Bitcoin address or even utilizing the micropayments feature from Lightning through the company's third-party channel to the merchant address.

The lightning network provides fast micropayment processing from invoices, however when is necessary to update such payments into the Bitcoin network then the current Bitcoin protocol is maintained, therefore at least 3 confirmations are necessary to occur before the withdrawal or micropayment transaction gets on-chain.

5.8 DISCUSSION

Lisk sidechain and Lightning network are different technologies, while the first is independent and have its own logic, exclusive blockchain, and custom-defined execution pace, the latter is an off-chain technology that runs on top of Bitcoin. The Lisk sidechain offers more flexibility in terms of customization characteristics of a business for example determining the fee costs on each transaction, size of a block, interval time of a block, and ability to send data messages and register them on-chain. Lightning cannot determine the fee costs of each transaction on Bitcoin and because of that it depends on the offer and demand of transaction space in a block in the Bitcoin network. The mining time of a block is defined as 10 minutes on Mainnet and Lightning requires at least 3 confirmations to update a transaction on-chain. After all experiments, it was concluded that both technologies are different and can be utilized in real scenarios.

Chapter 6

PRIVACY, COSTS, AND SCALABILITY BY THE DESIGN OF THE PROPOSED SOLUTION

Privacy of customer-sensitive data, the costs of a transaction, and the scalability of the solution are given by design. The correct utilization of the custom transaction type and sidechain configuration properties of the solution guarantees the cost and privacy demanded by our requirements. The following sections discuss privacy, cost, and performance with respect to the design of the solution.

6.1 VALIDATION OF PRIVACY BY THE DESIGN OF THE SOLUTION

Privacy of customer-sensitive data on the Food custom transaction type is guaranteed by the design of the custom transaction type itself and the characteristics of the blockchain. As stated in Chapter 4, the privacy of the Food Transaction type is guaranteed by the utilization of an encryption algorithm based on Edwards Curves 25519 and relies on the sender's private key, and receiver's public key for signing a transaction safely, also during the signature of the transaction the private key of the sender is not shared with anyone or network. Listing 6.1 illustrates how is displayed the sensitive fields of a transaction in the sidechain on public consultation. It is important to visualize the field senderPublicKey that belongs to the sender of the transaction, the recipientAddress field belongs to the receiver of the transaction, and the restaurantData and restaurantNonce have customer-sensitive encrypted data. During the sign of a transaction by the sender of it, behind the scenes its utilized the library created in this work Listing 6.2, the sensitive data information of the customer is signed as follows on Listing 4.1.

Listing 6.1 Food transaction tested on Scenario 7

```
{"moduleID":2000,"assetID":1040,"nonce":"0","fee":"0",
"senderPublicKey":"28c62cd9cb77ca3d356e99325c76b74167ee1b16fd80d9
b9b3c21658774c16e9",
"signatures":["ce0ae1de35edc18a2710857327ed19c08d67ea378984813b96
4129bd5a585add6c23fb9930c9a1b8329c8d066695e87012d8b18c6a1b8a354f5
af9690209a30b"],
"id":"42f39ad0fea0bbb7aa96ced9f617b32869d751efb7549befda92f6df619
4e3e4",
"asset":{
    "items":"[{\"name\":\"Black Pasta\",\"foodType\":1,
    \"quantity\":1,\"price\":0.1,
    \"observation\":\"\"}]",
    "price":"10000000",
"restaurantData":"97cd91c22e572ded03830048d41d73d26feb5e6076fb2
ac50baef0470cd5874f5d6536dd4117ddc5a2b4b1a8c02b9526b830c86ffe0b
0abcb3fb205a2e388fece7b9c4392d2a4ea5fe61b0f8f77733ed59f9db7284c
5d1795449f1177367bdef9542a9c66be4eefeab04668340f46ccac6a730463d
ca13",
"restaurantNonce":"6f0af9809ba09074c715b0921d9a099d293704c395bc
2d5e", "recipientAddress": "7028f454dc39d59368e040b1fa7b018d8d14f
894"}}]};
```

Listing 6.2 Creation of food transaction and transaction sign with helper library

```
async createFoodAssetAndSign(orderRequest, credential, restaurant){
    var recipientAddress =
cryptography.getAddressFromBase32Address(restaurant.address);

    const sender =
cryptography.getAddressAndPublicKeyFromPassphrase(
credential.passphrase);

    var accountNonce = await this.getAccountNonce(sender.address);

    var orderPrice = 0;

    var items = orderRequest.items;
    console.log("items:", typeof []);

    items.forEach(item =>{
        orderPrice += (item.price * item.quantity);
    });

    console.log("price", orderPrice);

    var restaurantData = cryptography.encryptMessageWithPassphrase(
        orderRequest.deliveryAddress
        .concat(' ***Field*** ')
        .concat(orderRequest.phone)
        .concat(' ***Field*** ')
        .concat(orderRequest.username),
        credential.passphrase,
        restaurant.publicKey);

    const tx = await transactions.signTransaction(
        schema,
        {
            moduleID: 2000,
            assetID: 1040,
            nonce: BigInt(accountNonce),
            fee: BigInt(0),
            senderPublicKey: sender.publicKey,
            asset: {
                items: JSON.stringify(orderRequest.items),
                price: BigInt(transactions.convertLSKToBeddows(
                    orderPrice.toString())),
                restaurantData: restaurantData.encryptedMessage,
                restaurantNonce: restaurantData.nonce,
                recipientAddress: recipientAddress
            },
        },
        Buffer.from(networkIdentifier, "hex"),
        credential.passphrase
    );

    return tx;
}
```

6.2 INCREASING THE SCALABILITY OF SOLUTION

A sidechain can be strategically configured to handle a specific number of users and devices, therefore maintaining an acceptable performance level, also other sidechains can be configured to handle an increase in the number of users and devices. Chapter 5 demonstrates a performance evaluation of the proposed sidechain of this work, its setups, and its usage recommendations. Therefore, we can think about increasing the number of sidechains proposed. This is done by parallelism which is allowed by the Lisk blockchain Technology, we can create as many sidechains as required which increases the processing of online food requests and increases the scalability of the solution as the proposed sidechains utilize the same LSK cryptocurrency. The mathematical representation of such an idea with the maximum number of food transactions in a block, 59 (evaluated in Chapter 5), can be expressed by $\sum_1^n *59$, where n represents the number of sidechains.

By default, in a sidechain users do not compete with users from other chains, and the proposed solution for restaurants allows the scale of more sidechains in case of an increase in the number of users and devices, therefore, maintaining an acceptable performance level. Despite that we did not experiment with the utilization of parallel sidechains, however, it is possible to verify such an argument by design.

6.3 LOOKING AT THE COSTS BY DESIGN

A sidechain has the benefit of specifying custom transaction types, furthermore, the transaction fee cost is configurable and it can be configured even to zero, such fact can be verified on Listing 6.1! On the other hand, it is necessary to configure the sidechain in nodes, such nodes can be handled by users, companies, or anyone interested in running a sidechain node. The reward mechanism property of a blockchain consensus mechanism is responsible for driving the system forward, hence a transaction fee also represents the reward a delegate account receives for each block forged. Hence, it is possible to configure a sidechain for custom cost, this permits costs to be reduced or increased based on the demand and planning of the sidechain owner.

6.4 ABOUT PERFORMANCE

Performance can be a concern in Bitcoin and Ethereum blockchains as already mentioned in this work. In a situation in which the number of transactions in such networks is high, there can occur an increase in waiting time from the moment of requesting a transaction until it gets executed and even a higher waiting time for the confirmation of the block containing such transaction. A sidechain can be strategically configured to handle a specific number of users and devices, therefore maintaining an acceptable performance level, also other sidechains can be configured to handle an increase in the number of users and devices. Chapter 5 demonstrates a performance evaluation of the proposed sidechain of this work, its setups, and its usage recommendations.

6.5 SUMMARY

This Chapter discussed aspects of privacy and costs that are guaranteed by the blockchain for the retail industry, reminding that Chapter 5 discussed aspects of performance and scalability in relation to the number of users. Based on Chapter 5 and Chapter 6 we could answer the research questions of this work.

Chapter 7

CONCLUSION

This chapter presents the conclusion of our work. Section 7.1 discusses the questions proposed in this thesis. Section 7.2 presents the limitations of our work. Section 7.3 discusses future work and Section 7.4 presents our final remarks on the blockchains of restaurants.

7.1 DISCUSSION

Here are presented the following answers to questions proposed in this thesis. They were proved by state-of-the-art research and experiment results that can be found in this work and on Github [1]:

- Is it possible to create a sidechain blockchain solution for the retail industry? Yes, as shown in Chapter 4, as it was observed in the experiments of Chapter 5, and the mentioned arguments in Chapter 6 it is possible to create a sidechain for the small retail industry

- Does it guarantee data privacy? Data privacy is guaranteed by the design of the solution as demonstrated in Chapter 4, and Chapter 6

- Does it have a low cost? Yes, the custom transaction type experimented in Chapter 5 was configured for a low cost. Chapter 6 verified the low cost of the solution by design. However, the number of nodes and how the solution is decentralized can determine the costs of the execution of the solution.

- Does it allow for a public audit of transactions? Yes, the audit of transactions is public, and is possible to track information from blockchain nodes, however, customer-sensitive data is private and protected as demonstrated in the previous Chapter.

[1] https://github.com/davilinfo/masterthesis

- Does it scale? Yes, the solution can afford an increase in the number of users and devices to a threshold. And more sidechain solutions can be deployed strategically to afford even more users and devices.
- What is its performance? As experimented with and observed in Chapter 5 the performance of the solution was measured. The solution offers reasonable performance for the small retail industry.
  - Does the increase in block size on the sidechain allow for a reduction in the time for forging a block on the restaurant sidechain? Yes, as the experiment results are shown in Table A.4 and Table A.6. However, It required more than one attempt to sync nodes and perform tests on Scenario 8 which it was observed stale blocks and nodes growing each chain separately. Upon Scenario 9 it was difficult too because of stale blocks and one of the 4 nodes started growing a chain separately because of the configuration utilized.
  - Does the increase of block size on the sidechain or reducing the time for forging a block provide better throughput when compared with the earlier version of the Restaurant sidechain (ALVES, 2021b)? Yes, as shown in the results of experiments in the previous Chapter, and results of the old version of Proof-Of-Concept (PoC) Lisk Restaurant (ALVES, 2021b), Table 5.7.

7.2 LIMITATIONS OF THE WORK

This work was not tested in a real restaurant during the writing of this dissertation. The source code of the solution is available in Github. For the proposed implementation, it was evaluated the sidechain and the integration library as explained in Chapters 5, and 6. Also, we did not perform a deep market analysis against other solutions, we did not verify the most commonly adopted solutions by the market, and we did not verify the quality of the implementation.

7.3 CONSIDERATION AND FUTURE WORKS

The Restaurant sidechain has a performance improvement when compared with its first version (ALVES, 2021b) in comparison with the capacity of Food transactions inside a block and speed to forge new blocks maintaining the capacity of transactions per block. Also, the created sidechain shows the possibility of blockchain application in the food industry respecting the data privacy of customers using cryptography based on private keys, public keys, nonce, and recipient addresses in a secure way for customers and restaurants. The comparison between the Lisk blockchain and Bitcoin, Multichain, and Ethereum blockchains shows the benefits of the sidechain solution proposed in aspects of lower computational costs and fees, block capacity, auditability, scalability, and encrypted storage on-chain. Finally, including the cryptocurrency that is supported globally expands the reachability of the proposed solution offering more competitive rates than credit card fees, especially, when utilized in a foreign country. For future work, an evolution of this solution can connect to food platforms and offer blockchain payment methods for its customers. An administration page to handle the inclusion of new restaurants can be explored. Decentralized and crypto exchanges can be utilized to facilitate the use of cryptocurrencies in the sidechains. A supply chain solution can connect and track the food ingredients from a farm to a restaurant table utilizing blockchain technology.

Another future work that is really interesting is regarding the block finalization time or block waiting time before the transaction is safely included in the blockchain without a risk of being reverted. In Bitcoin, such waiting time corresponds to 6 blocks created above the block that included a transaction, and such time is equivalent to 1 hour. In Ethereum, a finalization rule of a block was introduced in the document Casper The Friendly Ghost (BUTERIN; GRIFFITH, 2017) as per the creation of a checkpoint on every 100 blocks. In Lisk, during the evaluation of the sidechain the finalized property of a block was monitored, and it was observed a necessary waiting time equivalent to 26 minutes when the sidechain was configured with a block time of 10s and 13 minutes when the block time on the sidechain nodes was configured to 5s in Software Development Kit (SDK) 5. However, as stated in (HACKFELD, 2019) it is possible to finalize a block after 2 blocks creation above the forged block by utilizing a digital aggregation schema signature. Such a strategy was in development by Lisk, during the writing of this work, and it utilizes a BLS digital signature schema (JUSTIN, 2018). After the release of the Lisk 6.0 version, it included a faster finalization approach utilizing aggregation signatures and we performed an initial test finalization time of just 1 or 2 block(s) after a block creation, this is a big improvement allowing the wait of only 10 or 20 seconds for finalizing a block after a block creation, Listings A.7 and A.8. Before the release of the Lisk 6.0 version, we performed an initial and superficial analysis considering also a Ring digital aggregation signature schema (FANG et al., 2020) that requires the public keys from a group and requires only one private key of a group member without revealing the ownership of it. Such characteristics also can match with the Lisk Delegated Proof of Stake (DPoS) consensus mechanism which has a set of delegate accounts that forges blocks in the blockchain. Furthermore, the verification of such signature can be performed in polynomial time, in fact even faster as the delegates set on Lisk DPoS are known on each round and such information represents a constant value of 103 delegate accounts. Furthermore, it would be necessary for only 68 of 103 delegate accounts votes to achieve a majority when deciding on a block, which can be fast enough to achieve the goal of faster block finalization (RESEARCH, 2023).

7.4 THE BLOCKCHAINS OF RESTAURANTS

The blockchains of restaurants aim to be a solution that allows the integration of the food retail industry into blockchain respecting the privacy of customers with reduced transaction fees when compared to Bitcoin or Ethereum blockchains. A restaurant sidechain solution can be deployed into multiple sidechains utilizing the same cryptocurrency of Lisk Mainnet. It means that is possible to have multiple restaurant sidechain solutions per country, per city, per neighborhood, or necessity. Therefore increasing substantially scalability as explained in the previous Chapter in the Section Increase the Scalability of the solution.

7.4.1 Lisk Interoperability

While still in development at the moment of writing this thesis, the Lisk interoperability will allow the interoperability between Lisk sidechains, enhancing scalability in the network. It means that will be possible to transactions from one sidechain to another sidechain or from a sidechain to Lisk Mainnet. Version 6.0 from Lisk SDK was released and while still in the Betanet version, it brought interoperability features, and many other features including faster block finalization by the utilization of an aggregate signature schema.

7.4.2 Blockchain Interoperability

While still in development at the moment of writing this thesis Blockchain interoperability will allow the interoperability between different blockchains.

7.4.3 LSK TOKEN, Sidehchain TOKEN, BTC

LSK token can be utilized in the restaurant sidechain, it is accepted in several crypto exchanges, hence, it can be accessed in several different places in the world, and it is generated on each new block forged on the Lisk Mainnet network as a reward. The current reward is 1 LSK. Furthermore, a Sidechain token can be created if decided by the owner of a sidechain. BTC is the most known cryptocurrency in the world and it has a limited amount of generation on its network.

7.5 LAST WORDS

It is possible to find all source codes from the restaurant sidechain on GitHub.

https://github.com/davilinfo/MasterThesis

BIBLIOGRAPHY

AGRO, F. *Café brasileiro é exportado para o Japão com rastreabilidade blockchain pela primeira vez.* [S.l.], 2022. Last accessed on 18/05/2022. Disponível em: <https://forbes.com.br/forbesagro/2022/05/cafe-brasileiro-e-exportado-para-o-japao-com-rastreabilidade-blockchain-pela-primeira-vez/>.

AKBARI, E. et al. The impact of block parameters on the throughput and security of blockchains. In: *Proceedings of the 2020 The 2nd International Conference on Blockchain Technology.* New York, NY, USA: Association for Computing Machinery, 2020. (ICBCT'20), p. 13–18. ISBN 9781450377676. Disponível em: <https://doi.org/10.1145/3390566.3391673>.

ALESSANDRO, R. *Introducing Lisk Tree. Retrieved Sep 2020.* [S.l.], 2020. Disponível em: <https://lisk.io/blog/research/introducing-lisk-tree>.

ALESSANDRO, R. L. Lisk interoperability: Cross-chain interactions between homogeneous state machines. In: . [S.l.]: IEEE International Conference on Blockchain and Cryptocurrency, 2022.

ALHARBY, M.; MOORSEL, A. van. Blocksim: An extensible simulation tool for blockchain systems. *Frontiers in Blockchain*, v. 3, 06 2020.

ALVES, D. A strategy for mitigating denial of service attacks on nodes with delegate account of lisk blockchain. In: *Proceedings of the 2020 The 2nd International Conference on Blockchain Technology.* New York, NY, USA: Association for Computing Machinery, 2020. (ICBCT'20), p. 7–12. ISBN 9781450377676. Disponível em: <https://doi.org/10.1145/3390566.3391684>.

ALVES, D. The impact of denial-of-service attack for bitcoin miners, lisk forgers, and a mitigation strategy for lisk forgers. In: *Cybersecurity Threats with New Perspectives.* [S.l.]: IntechOpen, 2021.

ALVES, D. Proof-of-concept (POC) of restaurant's food requests in the lisk blockchain/sidechain. *Journal of Physics: Conference Series*, IOP Publishing, v. 1828, n. 1, p. 012110, feb 2021. Disponível em: <https://doi.org/10.1088/1742-6596/1828/1/012110>.

ALVES, D.; GREVE, F. Lisk restaurant sidechain, evolution from lisk sdk 2.3.8 to 5.0.3, evaluation of improvements in performance and security. In: *2021 The 3rd International Conference on Blockchain Technology.* New York, NY, USA: Association for Computing

Machinery, 2021. (ICBCT '21), p. 163–167. ISBN 9781450389624. Disponível em: <https://doi.org/10.1145/3460537.3460566>.

ARUNYADAV et al. Online food court payment system using blockchain technolgy. In: . [S.l.: s.n.], 2018.

BAYER, D.; HABER, S.; STORNETTA, W. Improving the efficiency and reliability of digital time-stamping. 09 1999.

BITCOIN. *Block hashing algorithm.* [S.l.], 2021. Last accessed on 20/09/2022. Disponível em: <https://en.bitcoin.it/wiki/Block_hashing_algorithm>.

BLOCKCHAIN. *BlockchainExplorer.* [S.l.], 2022. Last accessed on 30/05/2022. Disponível em: <https://www.blockchain.com/explorer?view=btc>.

BUTERIN, V.; GRIFFITH, V. Casper the friendly finality gadget. 10 2017.

CAMPBELL-VERDUYN, M. Bitcoin, crypto-coins, and global anti-money laundering governance. *Crime, Law and Social Change*, v. 69, 03 2018.

CHAUM, D. Blind signatures for untraceable payments. In: . [S.l.: s.n.], 1982. p. 199–203. ISBN 978-1-4757-0604-8.

DAMSGAARD, J. The real value of cryptocurrency. *Available at SSRN 4034312*, 2022. Last accessed 14/05/2022. Disponível em: <https://ssrn.com/abstract=4034312>.

DANIEL, B. et al. *Edwards curves 25519.* [S.l.], 2017. Last accessed on 01/12/2021. Disponível em: <https://en.wikipedia.org/wiki/EdDSA>.

DINH, T. et al. Blockbench: A framework for analyzing private blockchains. In: . [S.l.: s.n.], 2017. p. 1085–1100.

ESKANDARI, S.; CLARK, J.; HAMOU-LHADJ, A. Buy your coffee with bitcoin: Real-world deployment of a bitcoin point of sale terminal. In: . [S.l.]: 2016 Intl IEEE Conferences on Ubiquitous Intelligence & Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress (UIC/ATC/ScalCom/CBDCom/IoP/SmartWorld), 2016. p. 382–389.

ETHEREUM. *Ethereum documentation.* [S.l.], 2021. Last accessed on 02/12/21. Disponível em: <https://ethereum.org/>.

ETHERSCAN. *Etherscan.* [S.l.], 2022. Last accessed on 13/04/2022. Disponível em: <https://etherscan.io/gastracker/>.

ETHERSCAN. *Etherscan gas fee.* [S.l.], 2022. Last accessed on 23/04/2022 https://info.etherscan.com/what-is-gas-fee/. Disponível em: <https://info.etherscan.com/what-is-gas-fee/>.

FAN, C. et al. Performance evaluation of blockchain systems: A systematic survey. *IEEE Access*, v. 8, p. 126927–126950, 2020.

FANG, W. et al. Digital signature scheme for information non-repudiation in blockchain: a state of the art review. *EURASIP Journal on Wireless Communications and Networking*, v. 2020, 03 2020.

FERREIRA, E. et al. Uso de blockchain para privacidade e segurança em internet das coisas. In: ______. [S.l.: s.n.], 2017. p. 51. ISBN 9788576694106.

FOLHA, R. et al. Towards a novel business process model for food delivery services using blockchain technology. In: . [S.l.: s.n.], 2022.

GREENSPAN, G. *Multichain white paper*. [S.l.], 2014. Disponível em: <https://www.multichain.com/white-paper/>.

HACKFELD, J. A lightweight bft consensus protocol for blockchains. *arXiv preprint arXiv:1903.11434*, 2019.

IKER, A. *Introducing Lisk Dynamic Fee. Retrieved May 2020*. [S.l.], 2020. Disponível em: <https://lisk.io/blog/research/lisks-dynamic-fee-system>.

IKER, A. *New Lisk ID System*. [S.l.], 2020. Disponível em: <https://lisk.com/blog/research/new-lisk-id-system>.

JUSTIN, D. *Pragmatic signature aggregation with BLS*. [S.l.], 2018. Last accessed on 04/02/2023 https://ethresear.ch/t/pragmatic-signature-aggregation-with-bls/2105. Disponível em: <https://ethresear.ch/t/pragmatic-signature-aggregation-with-bls/2105>.

KHAN, N.; STATE, R. Lightning network: A comparative review of transaction fees and data analysis. In: PRIETO, J. et al. (Ed.). *Blockchain and Applications*. Cham: Springer International Publishing, 2020. p. 11–18. ISBN 978-3-030-23813-1.

Kim, S.-I.; Kim, S.-H. E-commerce payment model using blockchain. *Journal of Ambient Intelligence and Humanized Computing*, p. 1–13, 2020.

KLOMP, R.; BRACCIALI, A. On symbolic verification of bitcoin's script language. In: *Data Privacy Management, Cryptocurrencies and Blockchain Technology*. [S.l.]: Springer, 2018. p. 38–56.

LAMPORT, L. et al. Paxos made simple. *ACM Sigact News*, v. 32, n. 4, p. 18–25, 2001.

LAMPORT, L.; SHOSTAK, R.; PEASE, M. The byzantine generals problem. *ACM Trans. Program. Lang. Syst.*, v. 4, 02 2002.

LI, X. et al. From bitcoin to solana – innovating blockchain towards enterprise applications. In: LEE, K.; ZHANG, L.-J. (Ed.). *Blockchain – ICBC 2021*. Cham: Springer International Publishing, 2022. p. 74–100. ISBN 978-3-030-96527-3.

LIGHTNING. *Lightning Network documentation.* [S.l.], 2023. Last accessed on 11/05/2023. Disponível em: <https://lightning.readthedocs.io/>.

LIGHTNING. *Lightning Network HTLC documentation.* [S.l.], 2023. Last accessed on 15/05/2023. Disponível em: <https://docs.lightning.engineering/the-lightning-network/multihop-payments/hash-time-lock-contract-htlc>.

LISK. *Lisk SDK documentation. Retrieved Feb 2021.* [S.l.], 2021. Disponível em: <https://lisk.io/documentation/lisk-sdk>.

LISK. *Lisk website.* [S.l.], 2021. Last accessed on 01/12/21. Disponível em: <https://lisk.com/documentation/lisk-core>.

LISK. *Lisk HTTP API plugin.* [S.l.], 2022. Last accessed on 20/03/22. Disponível em: <https://lisk.com/documentation/lisk-sdk/references/lisk-framework/http-api-plugin.html#list-of-endpoints>.

LIU, J. et al. Bpds: A blockchain based privacy-preserving data sharing for electronic medical records. In: *2018 IEEE Global Communications Conference (GLOBECOM).* [S.l.: s.n.], 2018. p. 1–6.

LU, Y.; QI, Q.; CHEN, X. *A Framework of Transaction Packaging in High-throughput Blockchains.* 2023.

LUA, E. et al. A survey and comparison of peer-to-peer overlay network schemes. *Communications Surveys & Tutorials, IEEE*, v. 7, p. 72– 93, 04 2006.

MELO, C. A. S. d. Planejamento de infraestruturas computacionais para o provimento de serviços baseados em blockchain. Universidade Federal de Pernambuco, 2021.

MERKLE, R. Method of providing digital signatures. Stanford University, 1979. Disponível em: <https://patentimages.storage.googleapis.com/69/ab/d9/2ff9f94fada6ea/US4309569.pdf>.

MOORE, G. M. *BlockGrid: A Blockchain-Mediated Cyber-Physical Instructional Platform.* [S.l.], 2020.

NAKAMOTO, S. Bitcoin: A peer-to-peer electronic cash system. *Cryptography Mailing list at https://metzdowd.com*, 03 2009.

PAHL, C.; IOINI, N. E.; HELMER, S. A decision framework for blockchain platforms for iot and edge computing. In: . [S.l.: s.n.], 2018.

POON, J.; DRYJA, T. The bitcoin lightning network: Scalable off-chain instant payments. 2016.

RAJAN, R.; CAVALIERE, L. P. L.; PALLATHADKA, H. The concept of the cryptocurrency and the downfall of the banking sector in reflecting on the financial market. *Rentgenologiya i Radiologiya*, v. 60, p. 17–33, 05 2021.

RESEARCH, L. *Aggregate signature schema.* [S.l.], 2023. Last accessed on 04/02/2023 https://research.lisk.com/t/aggregate-signature-schema-to-aggregate-messages-in-child-block-of-b1/395/2. Disponível em: <https://research.lisk.com/t/aggregate-signature-schema-to-aggregate-messages-in-child-block-of-b1/395/2>.

RICOTTONE, A. *Lisk Merkle Tree.* [S.l.], 2021. Last accessed on 12/12/22. Disponível em: <https://github.com/LiskHQ/lips/blob/main/proposals/lip-0031.md>.

SAAD, M. et al. Exploring the attack surface of blockchain: A systematic overview. 04 2019.

SIMON, S. Brewer's cap theorem. *CS341 Distributed Information Systems, University of Basel (HS2012)*, 2000.

SINKOVIC, Z.; PRIBISALIć, L. Taxation of cryptocurrencies with income tax and corporate income tax. In: . [S.l.: s.n.], 2022. p. 1126–1131.

VASEK, M.; THORNTON, M.; MOORE, T. Empirical analysis of denial-of-service attacks in the bitcoin ecosystem. In: . [S.l.: s.n.], 2014. v. 8438, p. 57–71. ISBN 978-3-662-44773-4.

WIKIPEDIA. *Lightning network.* [S.l.], 2019. Last accessed on 11/05/2023. Disponível em: <https://en.wikipedia.org/wiki/Lightning_Network>.

WIKIPEDIA. *CAP theorem.* 2021. Last accessed on 26/08/21. Disponível em: <https://en.wikipedia.org/wiki/CAP_theorem>.

XIAO, Y. et al. A survey of distributed consensus protocols for blockchain networks. *IEEE Communications Surveys Tutorials*, v. 22, n. 2, p. 1432–1465, 2020.

YANG, D. et al. A review on scalability of blockchain. In: *Proceedings of the 2020 The 2nd International Conference on Blockchain Technology.* New York, NY, USA: Association for Computing Machinery, 2020. (ICBCT'20), p. 1–6. ISBN 9781450377676. Disponível em: <https://doi.org/10.1145/3390566.3391665>.

YERELI, A.; SAHIN, I. Cryptocurrencies and taxation. In: . [S.l.: s.n.], 2018.

ZOHAR, A. Bitcoin: Under the hood. *Commun. ACM*, Association for Computing Machinery, New York, NY, USA, v. 58, n. 9, p. 104–113, aug 2015. ISSN 0001-0782. Disponível em: <https://doi.org/10.1145/2701411>.

ZYSKIND, G. et al. Decentralizing privacy: Using blockchain to protect personal data. In: . [S.l.: s.n.], 2015. p. 180–184.

Appendix

A

APPENDIX

A.1 APPENDIX - RELATED WORK TABLE

Table A.1: Related works matrix

Id	title	datasets	year	first author
P001	Descentralizing Privacy: Using Blockchain to Protect Personal Data	1 - IEEE CS Security and Privacy Workshops	2015	Zyskind
P002	A lightweight BFT consensus protocol for blockchains	Arxiv	2019	Jan Hackfeld
P003	The impact of block parameters on the throughput and security of blockchains	ACM	2020	AKBARI
P004	Performance Evaluation of Blockchain Systems: A Systematic Survey	IEEE	2020	C. Fan
P005	Blockchain e a Revolução do Consenso sob Demanda	SBC	2018	Fabíola Greve
P006	Blockchain a Technology Overview	NIST Draft NISTIR 8202, available on https://csrc.nist.gov	2018	Dylan Yaga
P007	Blockchains and Smart Contracts for the Internet of Things	IEEE	2016	C. Konstantinos
P008	A Decision Framework for Blockchain Platforms for IoT and Edge Computing	1-3rd International Conference on Internet of Things, Big Data and Security	2018	P. Claus

Table A.1: Related works matrix

Id	title	datasets	year	first author
P009	Proof-of-concept (POC) of restaurant's food requests in the lisk blockchain/sidechain	IOP Publishing	2021	Davi Alves
P010	Online Food Court Payment System using Blockchain Technology	IEEE	2018	Yadav D.
P011	Buy your coffee with bitcoin: Real-world deployment of a bitcoin point of sale terminal	IEEE	2016	Eskandary S.
P012	E-commerce payment model using blockchain	AIHC	2020	Kim S.
P013	Towards a novel business process model for food delivery services using blockchain technology	AMCIS	2022	Rodrigo F.
P014	BLOCKGRID: A BLOCKCHAIN-MEDIATED CYBER-PHYSICAL INSTRUCTIONAL PLATFORM	ACM	2020	Gregory M.
P015	From Bitcoin to Solana Innovating Blockchain Towards Enterprise Applications	Springer	2022	Li X.
P016	Paxos made simple	ACM	2021	Leslie Lamport
P017	BPDS: A blockchain based privacy-preserving data sharing for electronic medical records	IEEE	2018	Liu J.
P018	On symbolic verification of bitcoin's script language	Springer	2018	klomp R.
P019	Edwards curves 25519		2021	Daniel B.
P020	The real value of cryptocurrency		2022	Damsgaar
P021	Improving the efficiency and reliability of digital time-stamping	Springer	1999	Bayer D.

Table A.1: Related works matrix

Id	title	datasets	year	first author
P022	Lisk interoperability: Cross-chain interactions between homogeneous state machines	IEEE	2022	Alessandro R.
P023	A strategy for mitigating denial of service attacks on nodes with delegate account of lisk blockchain	ACM	2020	Davi Alves
P024	Multichain white paper		2014	Greenspan
P025	Lisk restaurant sidechain, evolution from lisk sdk 2.3.8 to 5.0.3, evaluation of improvements in performance and security	ACM	2021	Davi Alves
P026	The impact of denial-of-service attack for bitcoin miners, lisk forgers, and a mitigation strategy for lisk forgers	INTECHOPEN	2021	Davi Alves
P027	Lightning network: A comparative review of transaction fees and data analysis	Springer	2020	Khan N.
P028	Method of providing digital signatures	Stanford	1979	R. Merkle
P029	Bitcoin: A peer-to-peer electronic cash system.	Cryptography Mailing list at https://metzdowd.com	2009	S. Nakamoto
P030	The bitcoin lightning network: Scalable off-chain instant payments		2016	Joseph P.
P031	The concept of cryptocurrency and the downfall of the banking sector in reflecting on the financial market	Rentgenologiya i Radiologiya	2021	Rajan R.
P032	Exploring the attack surface of blockchain: A systematic overview.		2019	M. Saad

Table A.1: Related works matrix

Id	title	datasets	year	first author
P033	Empirical analysis of denial-of-service attacks in the bitcoin ecosystem	International Conference on Financial Cryptography and Data Security	2014	M. Vasek
P034	A survey of distributed consensus protocols for blockchain networks	IEEE	2020	Xiao Y.
P035	Bitcoin: Under the hood	ACM	2020	Zohar A.
P036	BLOCKBENCH: A Framework for Analyzing Private Blockchains	ACM	2017	T. Tien
P037	Dos and Don'ts in Blockchain Research and Development	ACM	2020	Wenbing Zhao
P038	BITCOIN, CRYPTO-COINS, AND GLOBAL ANTI-MONEY LAUNDERING GOVERNANCE	Springer	2018	V. Campbell
P039	Taxation of Cryptocurrencies with Income Tax and Corporate Income Tax		2022	Sinkovic Z.
P040	Cryptocurrencies and Taxation	5th International Annual Meeting of Sosyoekonomi Society	2018	Ahmet
P041	Brewer's cap theorem	Distributed Information Systems	2000	Simon B.
P042	A survey and comparison of peer-to-peer overlay network schemes	Communications Surveys & Tutorials, IEEE	2006	Lua E.
P043	The Byzantine Generals Problem	ACM	2002	Leslie Lamport
P044	A Review on Scalability of Blockchain	ACM	2020	Di Yang
P045	Planejamento de infraestruturas computacionais para o provimento de serviços baseados em blockchain	UFPE	2021	Melo C.
P046	A Framework of Transaction Packaging in High-throughput Blockchains	arXiv	2023	Yuxuan Lu

A.2 APPENDIX - LISTINGS OF TRANSACTIONS

Listing A.1 Helper method for creating a Food transaction

```
async createFoodAssetAndSign(orderRequest, credential, restaurant){
    var recipientAddress =
cryptography.getAddressFromBase32Address(restaurant.address);

    const sender =
cryptography.getAddressAndPublicKeyFromPassphrase(
credential.passphrase);

    var accountNonce = await this.getAccountNonce(sender.address);
    var orderPrice = 0;

    var items = orderRequest.items;
    console.log("items :", typeof []);

    items.forEach(item =>{
        orderPrice += (item.price * item.quantity);
    });

    console.log("price", orderPrice);

    var restaurantData = cryptography.encryptMessageWithPassphrase(
    orderRequest.deliveryAddress
    .concat(' ***Field*** ')
    .concat(orderRequest.phone)
    .concat(' ***Field*** ')
    .concat(orderRequest.username),
    credential.passphrase,
    restaurant.publicKey);

    const tx = await transactions.signTransaction(
    schema,
    {
        moduleID: 2000,
        assetID: 1040,
        nonce: BigInt(accountNonce),
        fee: BigInt(0),
        senderPublicKey: sender.publicKey,
        asset: {
            items: JSON.stringify(orderRequest.items),
            price:
    BigInt(transactions.convertLSKToBeddows(orderPrice.toString())),
            restaurantData: restaurantData.encryptedMessage,
            restaurantNonce: restaurantData.nonce,
            recipientAddress: recipientAddress
        },
    },
    Buffer.from(networkIdentifier, "hex"),
    credential.passphrase);

    return tx;
}
```

Listing A.2 FoodTransaction validate method

```
validate({asset}){
    var items = JSON.parse(asset.items);
    for (var index=0; index < items.length; index ++){
        if (!items[index].name || typeof items[index].name !==
        'string' || items[index].name.length > 200){
            throw new Error(
                    'Invalid "asset.items[index].name" defined on
                    transaction: A string value no longer
                    than 200 characters. index:
                    '.concat(index.toString()).concat('
                    ').concat(asset.items[index].name));
        }

        if (!items[index].foodType || items[index].foodType <= 0){
            throw new Error(
                    'Invalid "asset.items[index].foodType"
                    defined on transaction:
                    A value bigger than 0. index:
                    '.concat(index.toString()).concat('
                    ').concat(asset.items[index].foodType));
        }

        if (!items[index].quantity || items[index].quantity < 0){
            throw new Error(
                    'Invalid "asset.items[index].quantity"
                    defined on transaction:
                    A value bigger than 0. index:
                    '.concat(index.toString()).concat('
                    ').concat(asset.items[index].quantity));
        }
        if (!items[index].price || items[index].price < 0){
            throw new Error(
                    'Invalid "asset.items[index].price"
                    defined on transaction:
                    A value bigger than 0. index:
                    '.concat(index.toString()).concat('
                    ').concat(asset.items[index].price));
        }
    }

    if (!asset.restaurantData ||
    asset.restaurantData.length === 0){
        throw new Error(
            'Invalid "restaurantData" defined on transaction:
                Not empty');
    }

    if (!asset.restaurantNonce ||
    asset.restaurantNonce.length === 0){
        throw new Error(
            'Invalid "restaurantNonce" defined on transaction:
                Not empty');
    }
}
```

Listing A.3 Food transaction with several meals

```
{"data":
[{"header":
{"version":2,"timestamp":1684709106,
"height":86116,
"previousBlockID":"1a537165a014f3dc3829e294ff...",
"transactionRoot":"e154771656068df0654ee572...",
"generatorPublicKey":"0c95295ca781af27c81e66143f7...",
"reward":"500000000",
"asset":{"maxHeightPreviouslyForged":86084,"maxHeightPrevoted":0,
"seedReveal":"bf0874ad068d23869e16be4a639c5859"},
"signature":"069377d04cd39a6eb4a801c6b02e3357...",
"id":"d960240fafd786d5253f4e9429061ac7f69375835008d13a68657ea6f90
c0c70"},
"payload":[{"moduleID":2000,"assetID":1040,"nonce":"13","fee":"0",
"senderPublicKey":"3f31f1e1e79209e3898c46a30770c7...",
"asset":{
"items":"[
    {"name":"Osso Buco","foodType":8,"quantity":1,
        "price":25.89,"observation":""},
    {"name":"Filetto di Manzo","foodType":7,"quantity":1,
        "price":26,"observation":""},
    {"name":"Gelato","foodType":5,"quantity":2,
        "price":4,
    "observation":""},
    {"name":"Bottle of water","foodType":14,"quantity":2,
        "price":2,"observation":""}]
","price":"6389000000",
"restaurantData":"10f9a36aaafdcb364b3ebe17e12f99bd03f49939e620937142...",
"restaurantNonce":"69e67e9b1e58afb596a5eda62608f15b4bbd025e22771524",
"recipientAddress":"7028f454dc39d59368e040b1fa7b018d8d14f894"},
"signatures":["7e31bc92f6294c25c072ee2c0578a39a5782549bb467f9
..."],
"id":"27faabfaea7c9c8bafc677519dac13c3dd9216b21bfad724857057a2d1893df2"
}]}],"meta":{}}
```

Listing A.4 MenuTransaction validate method

```
validate({asset}){
    if (!asset.items){
        throw new Error(
            'Restaurant menu should include food and/or
            beverages. Please include at least some item:
            "asset.items"');
    }
    var items = JSON.parse(asset.items);
    for (var index=0; index < items.length; index ++){

        if (!items[index].name ||
        typeof items[index].name !== 'string' ||
        items[index].name.length > 200){
            throw  new Error(
                'Invalid "name" defined on transaction
                "asset.items[index].name .
                Should be included a string value no longer than
                200 characters"'
                );
        }

        if (!items[index].description ||
        typeof items[index].description !== 'string' ||
        items[index].description.length > 2000){
            throw new Error(
                'Invalid "description" defined on transaction
                "asset.items[index].description.
                Should be included a string value no longer than
                2000 characters"'
            );
        }

        if (!items[index].price ||
        items[index].price < 0 ){
            throw new Error(
                'Invalid "price" defined on transaction
                "asset.items[index].price" .
                A value equal or bigger than 0'
                );
        }

        if (!items[index].discount ||
        items[index].discount < 0 ){
            throw new Error(
                'Invalid "asset.items[index].discount" defined on
                transaction . A value equal or bigger than 0'
            );
        }

        if (!items[index].type){
            throw new Error(
                'Invalid "asset.items[index].type" defined on
                transaction . A number bigger than 0'
            );
        }

        if (!items[index].category){
            throw new Error(
                'Invalid "asset.items[index].category"
                defined on transaction . A number bigger than 0'
            );
        }

        if (!items[index].img){
            throw new Error(
                'Invalid "asset.items[index].img" defined on
                transaction .
                A string http address of the food image'
            );
        }
    }
}
```

Listing A.5 ProfileTransaction validate method

```
validate({asset}){

    if (!asset.name ||
    typeof asset.name !== 'string' ||
    asset.name.length > 200){
        throw new Error(
                'Invalid "asset.name" defined on transaction:
                A string value no longer than 200 characters');
    }

    if (!asset.clientData || asset.clientData.length === 0){
        throw new Error(
                'Invalid "clientData" defined on transaction:
                Not empty');
    }

    if (!asset.clientNonce || asset.clientNonce.length === 0){
        throw new Error(
                'Invalid "clientNonce" defined on transaction:
                Not empty');
    }
}
```

Listing A.6 NewsTransaction validate method

```
validate({asset}){
    if (!asset.items){
        throw new Error(
            'Please include at least a news: "asset.items"');
    }
    var items = JSON.parse(asset.items);
    for (var index=0; index < items.length; index ++){

        if (!items[index].title ||
        typeof items[index].title !== 'string' ||
        items[index].title.length > 50){
            throw  new Error(
                'Invalid "name" defined on transaction
                "asset.items[index].name .
                Should be included a string value
                no longer than 50 characters"'
            );
        }

        if (!items[index].description ||
        typeof items[index].description !== 'string' ||
        items[index].description.length > 160){
            throw new Error(
                'Invalid "description" defined on transaction
                "asset.items[index].description.
                Should be included a string value
                no longer than 160 characters"'
            );
        }

        if (!items[index].text ||
        items[index].text.length > 2000 ){
            throw new Error(
                'Invalid "text" defined on transaction
                "asset.items[index].text" .
                Should be included a string value
                no longer than 2000 characters'
            );
        }
    }
}
```

Listing A.7 SDK 6.0 block finalization Channel

```
{"header":{"version":2,"timestamp":1688375550,"height":19240,
"previousBlockID":"fd8ad1dd0f36e7441c3b390072a6
3002baf0e1eee3cc1b86d30e99ed930fa9bc",
"stateRoot":"037533cdd9472e14281be1d0340e3878bd
cd0d31b174929208d49d1637506625",
"assetRoot":"1df5a979a6c2205634dba768b9581ceb68
6672478282fae6dc0e8aa8c1daf9ae",
"eventRoot":"372d5e61c9fc00da9a86d546d2788c408d
3e90e994077d7b9d8ebc2794d0fe67",
"transactionRoot":"730d4eff4599d586c97b3f6ec5bb
d843e47310852a606278cb0a7a23ef12b2ac",
"validatorsHash":"470c7d62d9d2b546efbd5363f9f7f
86d8bfb68627e4813fb824943d7d3fd5c45",
"aggregateCommit":{"height":19238,"aggregationBits":"01",
"certificateSignature":"99ce3561437d8ae1f75b7d0a
a8f9ce457b35ceed1f06dbac77ec2c6f5093663a4e1b2ae2
859814f0b4909b3a966ab6db1458d7e3eb7992ff537f59ca
b37276cd300eb24185143023713bbf86e4abb16ab729a271
da822a184869b76ae63bc55e"},
"generatorAddress":"lskzpzzros5qe5mrstgqr33h8vrsyrmaddart6pqy",
"maxHeightPrevoted":19239,
"maxHeightGenerated":19239,"impliesMaxPrevotes":true,
"signature":"81788c81466d4767f5833c46be15c5354a1
b9322da49e239cdc56e81ba914f98d8e7d45501f19192ee3
eff96d007a36383f15c66160d60f33218f8e05b911f05",
"id":"144b42dea6abca3a861c6053dd869d881012e40ef7fa
17dd7af5a3eab3161e87"},
"transactions":[{"module":"hello","command":"createHello",
"params":{"message":"First message"},"nonce":"0","fee":"1000000",
"senderPublicKey":
"e64d0d5214be8b38bd67ef50685624dd73b453c2661f4ca680d914be701c3f66",
"signatures":["7f67e700b3689ecefbd458162dc5af370a25f6c08fd6aafb
be252d4b1bbc864815cfa5734833565013ddb45f320f31116f
213012b82f59c336b7282327ce0208"],"id":"0593044612d
2391558d7056944d7af533aabdead0fa65f48fcbede9469db6807"}],
"assets":[{"module":"random",
"data":{"seedReveal":"e420befe20443ec56f3c8f19a86090f6"}}]}
```

Listing A.8 SDK 6.0 block finalization

```
{"version":"0.1.0","networkVersion":"1.0",
"chainID":"0000abcd",
"lastBlockID":
"a97f7c343a23d183598e1c25a089647aecd277d941d5d324e38561589b479494",
"height":22266,
"finalizedHeight":22265,
"syncing":false,
"unconfirmedTransactions":0,"genesisHeight":0,
"genesis":{"block":{"fromFile":"./config/genesis_block.blob"},
"blockTime":10,
"bftBatchSize":103,
"maxTransactionsSize":15360,
"minimumCertifyHeight":1,"chainID":"0000abcd"},
"network":{"version":"1.0","port":7667,"seedPeers":[]}}
```

A.3 APPENDIX - TABLES OF RESULTS OF EXPERIMENTAL STUDIES

Table A.2 Scenario 1 - Evaluation of the number of Food Asset transactions included in a block and waiting time by a node that received transactions. Each node was tested utilizing the library proposed in this paper. 10s block interval, 30kb block size, 4 nodes located in distinct data centers

Users	maximum nº of Tx/block	Waiting time
64	59	20 seconds
110	59	20 seconds
150	59	30 seconds
200	59	40 seconds

Table A.3 Scenario 2 - Evaluation of the maximum number of Food Asset transactions included in a block and maximum waiting time by a node that received the transactions. Each node was tested by time utilizing the spam.js script. 10s block interval, 25kb block size

Users	maximum nº of Tx/block	Waiting time
64	49	20 seconds
110	49	20 seconds
150	49	40 seconds
200	49	60 seconds

Table A.4 Scenario 3 - Evaluation of the maximum number of Food Asset transactions included in a block and maximum waiting time by a node that received the transactions. Each node was tested by time utilizing the spam.js script. 10s block interval, 20kb block size

Users	maximum nº of Tx/block	Waiting time
64	39	20 seconds
110	39	30 seconds
150	39	50 seconds
200	39	60 seconds

Table A.5 Scenario 4 - Evaluation of the maximum number of Food Asset transactions included in a block and maximum waiting time by a node that received the transactions. Each node was tested by time utilizing the spam.js script. 10s block interval, 15kb block size

Users	maximum nº of Tx/block	Waiting time
64	30	30 seconds
110	30	40 seconds
150	30	50 seconds
200	30	70 seconds

Table A.6 Scenario 5 - Evaluation of the maximum number of Food Asset transactions included in a block and maximum waiting time by a node that received the transactions. Each node was tested by time utilizing the spam.js script. 5s block interval, 20kb block size

Users	maximum nº of Tx/block	Waiting time
64	39	10 seconds
110	39	15 seconds
150	39	20 seconds
200	39	30 seconds

Table A.7 Scenario 6 - Evaluation of the maximum number of Food Asset transactions included in a block and maximum waiting time by a node that received the transactions. Each node was tested by time utilizing a script. 5 seconds block interval, 30kb block size

Users	maximum nº of Tx/block	Waiting time
64	59	10 seconds
110	59	10 seconds
150	59	15 seconds
200	59	20 seconds

Table A.8 Scenario 8 - Evaluation of the maximum number of Food Asset transactions included in a block and maximum waiting time by a node that received the transactions. Each node was tested by time utilizing a script. 5 seconds block interval, 25kb block size

Users	maximum nº of Tx/block	Waiting time
64	49	10 seconds
110	49	20 seconds
150	49	20 seconds
200	49	25 seconds

Table A.9 Scenario 9 - Evaluation of the maximum number of Food Asset transactions included in a block and maximum waiting time by a node that received the transactions. Each node was tested by time utilizing a script. 5 seconds block interval, 15kb block size

Users	maximum nº of Tx/block	Waiting time
64	30	15 seconds
110	30	20 seconds
150	30	25 seconds
200	30	35 seconds

A.4 APPENDIX - LISTINGS OF INVOICES AND PAYMENTS FROM THE EXPERIMENTS

Listing A.9 Lightning Experiment 01 - invoices generated by Node 1 and paid by Node 2 utilizing Faucet node as router

```
[
    {
        "label": "invoice01...",
        "bolt11": "lntb20n1pj8avzhsp5735fjnt0dzjsur5xh8ansrj7az5v5
        wp42sxrhh0ecucn3ecrz9cspp54574fannzw3vetxtxjkc9pv3aq5rh5v
        qzmmax695sq9pwjnxxqrsdquv3jhxcmjd9c8g6t0dcsxyun9v9jqxqraf
        scqp2rzjqfcxsh9gr28y6ngphmk90q05ejfydpq89tjjc5rl36lfmtcv4
        24hkffd7qqqqdcqqyqqqqlgqqqqqqgq2q9qyysgqq4qp3hkhft75hnxlz
        g33ntpk6dlad67fe49vl45t8ej5jkmnln6hk98mvke58yeanq4au4elgv
        up8dpelzprzx743vjr52yyhtp7tucqwrhvjf",
        "payment_hash": "ad3d54f67313a2ccaccb34ad828591e8283bd1801
        6f7d368b4800a174a663007",
        "msatoshi": 2000,
        "amount_msat": "2000msat",
        "status": "paid",
        "pay_index": 11,
    "msatoshi_received": 2000,
    "amount_received_msat": "2000msat",
    "paid_at": 1686024302,
    "payment_preimage": "02729e9c64ca2be0baa40b54f4d543ace0c8b
    968a0234f112af9e0c0f0fab8f5",
    "description": "description bread",
    "expires_at": 1686054279
  },
  {
    "label": "invoice02..",
    "bolt11": "lntb20n1pj8avxasp5g9nw36536dcc6793ffy5utg3qw6mf88
    w3736nh6s6lgmjktqgnpspp50vawucmlty7eez9axr8gzcf7lgark6qas
    vruc05dwrpr8x2wwwyqdquv3jhxcmjd9c8g6t0dcsxyun9v9jqxqrafsc
    qp2rzjqfcxsh9gr28y6ngphmk90q05ejfydpq89tjjc5rl36lfmtcv424
    hkffd7qqqqdcqqyqqqqlgqqqqqqgq2q9qyysgq6zenez6ulmcdmu7r85m
    c09jl5m5erfwpquyyulgm8s3r7975597scahjjlnrye85f0c83px6ahpn
    75qphwefmwp6cjhfp26spcdcaqqq8vezk9",
    "payment_hash": "7b3aee637f593d9c88bd30ce81613efa3a3b681d8307
    cc3e8d70c233994e7388",
    "msatoshi": 2000,
    "amount_msat": "2000msat",
    "status": "paid",
    "pay_index": 12,
    "msatoshi_received": 2000,
    "amount_received_msat": "2000msat",
    "paid_at": 304,
    "payment_preimage": "dbd59db64358e11c4ffddb6345127bb1f9c1113f
    4689af3783d0a5f3e722fa95",
    "description": "description bread",
    "expires_at": 1686054413
  },
  {
    "label": "invoice03..",
    "bolt11": "lntb20n1pj8avxasp5dgpdcpvs9gw9pd2nqarafsj3z4sc8jyrx
    we9jgjtt2362m3u27sqpp5083jrueve5uee34820ptfkec7j80lf2mek47
    wrgutcnk2vveywvsdquv3jhxcmjd9c8g6t0dcsxyun9v9jqxqrafscqp2r
    zjqfcxsh9gr28y6ngphmk90q05ejfydpq89tjjc5rl36lfmtcv424hkffd
    7qqqqdcqqyqqqqlgqqqqqqgq2q9qyysgquteae8h94yd52z7m8g49aya65
    4rfgucljsgqwn9q9wrpc9t3kc3szdnk00cveywakkgrpc4530kcrpzl5cf
    ar93szff0l262e57320gqvpfg8k",
    "payment_hash": "79e321f32ccd399cc6a753c2b4db38f48effa55bcdab
    e70d1c5e276531992399",
    "msatoshi": 2000,
    "amount_msat": "2000msat",
      "status": "paid",
      "pay_index": 13,
      "msatoshi_received": 2000,
      "amount_received_msat": "2000msat",
      "paid_at": 308,
      "payment_preimage": "27b0673ff84e8a3c9f47321b2fb4ce7d787
      641eca358f8dc3327a872d1311b43",
      "description": "description bread",
      "expires_at": 1686054413
   },
   {
      "label": "invoice04..",
      "bolt11": "lntb20n1pj8avxasp5mz7h9tnfp67nwe72vtt8swrzfscqy
      jaf2c5nff4gh3zf8v6nherqpp5wv86mgm5r4kyymztlmx30yma84vg8z7dj
      y8gsqnmtf3v4yajv0nsdquv3jhxcmjd9c8g6t0dcsxyun9v9jqxqrafscqp
      2rzjqfcxsh9gr28y6ngphmk90q05ejfydpq89tjjc5rl36lfmtcv424hkffd
      7qqqqdcqqyqqqqlgqqqqqqgq2q9qyysgq4lsjk7gvnz50hvtsclrz6u57g7r
      e4plal8ch3s2u5qjasjrpyky4glkv5rtm3txkt79apjjf99alpewdyhr7pc0
      8prd5uph73c6vqgqq9skjp8",
      "payment_hash": "730fada3741d6c426c4bfecd17937d3d58838bcd
      910e88027b5a62ca93b263e7",
      "msatoshi": 2000,
      "amount_msat": "2000msat",
      "status": "paid",
      "pay_index": 14,
      "msatoshi_received": 2000,
      "amount_received_msat": "2000msat",
      "paid_at": 311,
      "payment_preimage": "2b81a8cbcee04a5942e83b002bc970c06a845b0
      df9f5b3eca18aee2d102b93b8",
      "description": "description bread",
      "expires_at": 1686054413
   },
   {
      "label": "invoice05..",
      "bolt11": "lntb20n1pj8avxasp5qa8d0kjm7kgdnypn08nvzjgu4yryyxy
      nns8nftdzyeqsfv9nhdnqpp5m2w87hred99cmlymez9z3g9v58s38ynfj
      dph8v3geasatvk6w34qdquv3jhxcmjd9c8g6t0dcsxyun9v9jqxqrafsc
      qp2rzjqfcxsh9gr28y6ngphmk90q05ejfydpq89tjjc5rl36lfmtcv424
      hkffd7qqqqdcqqyqqqqlgqqqqqqgq2q9qyysgqhwtgvcmjtrq2p87vjpz
      66rju42yg3sxcnajq0jlyprf3qa4rhgvkkmv9pgldw5gdpf6mtytzh34a

      7pgkdllxykujt7x89aw5xwaztmgp2jnaa6",
      "payment_hash": "da9c7f5c79694b8dfc9bc88a28a0aca1e11392699
```

```
        34373b228cf61d5b2da746a",
        "msatoshi": 2000,
        "amount_msat": "2000msat",
        "status": "paid",
        "pay_index": 15,
        "msatoshi_received": 2000,
        "amount_received_msat": "2000msat",
        "paid_at": 315,
        "payment_preimage": "da2888594fc6c7067e6db9b3847da845ca4f5f24
        e246ae83a85fc4affa088081",
        "description": "description bread",
        "expires_at": 1686054413
    },
    {
        "label": "invoice06..",
        "bolt11": "lntb20n1pj8avxasp580pj8yeghuqqf3ufarljy3k2xzgtll
         lzqsvzsr07apx0cxalnthqpp5a3cs033330k79pf6u68svt0h5wr26n9qm06
         746sqd7yg98vdc2pqdquv3jhxcmjd9c8g6t0dcsxyun9v9jqxqrafscqp2rzj
         qfcxsh9gr28y6ngphmk90q05ejfydpq89tjjc5rl36lfmtcv424hkffd7qqqq
        dcqqyqqqlgqqqqqqgq2q9qyysgqfl3qmv28eg688yvm994dfmpckgqtxud
        v36u5cnc8vqp42xrnvlmnevks8g8yzk3rx3hd8epsa09nkjd3aeuuj4scuf
        56sjlrjj6cpxspts0yfp",
        "payment_hash": "ec7107c6318bede2853ae68f062df7a386ad4ca0dbf
        5eaea006f88829d8dc282",
        "msatoshi": 2000,
        "amount_msat": "2000msat",
        "status": "paid",
        "pay_index": 16,
        "msatoshi_received": 2000,
        "amount_received_msat": "2000msat",
        "paid_at": 318,
        "payment_preimage": "d28789e7614b407e642830d4b60361e9120f9519a4b
        69876fcb4b0454beb4d7a",
        "description": "description bread",
        "expires_at": 1686054413
    },
    {
        "label": "invoice07..",
        "bolt11": "lntb20n1pj8avxasp5lm3n230kk2fl8h48ldh5gj2nq0xc7j
        djesesz2hvy5jraz6q0fxqpp5sm4rakfashnrdv57fhxnfp7hfup7hud46
        3nk2hhwzen88fmtgh2sdquv3jhxcmjd9c8g6t0dcsxyun9v9jqxqrafscqp
        2rzjqfcxsh9gr28y6ngphmk90q05ejfydpq89tjjc5rl36lfmtcv424h
        kffd7qqqqdcqqyqqqlgqqqqqqgq2q9qyysgqlq69a287d932cvexleg
        l2tyshaa8a6cau4yn0ll963n6ptugtqf993r65j5kz80nsda2e6ryw4c
        l2kxjlflgvjpwsyykgdur00kpn8gqm7awzx",
        "payment_hash": "86ea3ed93d85e636b29e4dcd3487d74f03ebf1b5d46
        7655eee166673a76b45d5",
        "msatoshi": 2000,
        "amount_msat": "2000msat",
        "status": "paid",
        "pay_index": 17,
        "msatoshi_received": 2000,
        "amount_received_msat": "2000msat",
        "paid_at": 321,
        "payment_preimage": "601bcf260f25e01c1ec5f347557590a387d1a6cc
        70b048a921e30bfb2651eb20",
        "description": "description bread",
        "expires_at": 1686054413
    },
    {
        "label": "invoice08..",
        "bolt11": "lntb20n1pj8avx7sp50sxp79a6y87khzp0aqyfxxhf5gz8m
        e4lxj9uhtcncka7elpzrwjqpp58dg6sncz3eux3jzuhqwfy95q00mgt079
        7vr0z8z9eawvj9mjsmtqdquv3jhxcmjd9c8g6t0dcsxyun9v9jqxqrafs
        cqp2rzjqfcxsh9gr28y6ngphmk90q05ejfydpq89tjjc5rl36lfmtcv42
        4hkffd7qqqqdcqqyqqqlgqqqqqqgq2q9qyysgqar4rmtjtzccvekpffk
        f8npxvy5lp2l3v6kzfd9lmzfupqj4xfacs9wmcerpkp376qndfu54upmy

        sk5e3dym47ad7srfh0wxmcewj0hspqufr46",
        "payment_hash": "3b51a84f028e7868c85cb81c9216807bf685b
        fc5f306f11c45cf5cc9177286d6",
        "msatoshi": 2000,
        "amount_msat": "2000msat",
        "status": "paid",
        "pay_index": 18,
        "msatoshi_received": 2000,
        "amount_received_msat": "2000msat",
        "paid_at": 325,
        "payment_preimage": "63e51dd136eda6a1ddd0c57ba75fe6d062df7
        447e9f39ae2e06cf4c48b958384",
        "description": "description bread",
        "expires_at": 1686054414
    },
    {
        "label": "invoice09..",
        "bolt11": "lntb20n1pj8avx7sp56vc53dzt3ra6zrrulcrfw4d8egmu
        q29hn5vg47u2646099mfv98qpp5axdr9sntnj5pz3k9hl4azyqahgtewjz
        u98y6f3sxhvkeupmgzvuqdquv3jhxcmjd9c8g6t0dcsxyun9v9jqxqr
        afscqp2rzjqfcxsh9gr28y6ngphmk90q05ejfydpq89tjjc5rl36lfm
        tcv424hkffd7qqqqdcqqyqqqqlgqqqqqqgq2q9qyysgq93qz69ptsld
        ddh5p6e7nygeru44szn69hf2ka8v8kzwh02qnjespm6zn37hu2mu6yr

        0vfa3njjqjvtazqv7l23u8mzuyqjffaymewasp5q2atm ,
        "payment_hash": "e99a32c26b9ca81146c5bfebd1101dba1797485c29c
        9a4c606bb2d9e07681338",
        "msatoshi": 2000,
        "amount_msat": "2000msat",
        "status": "paid",
        "pay_index": 19,
        "msatoshi_received": 2000,
        "amount_received_msat": "2000msat",
        "paid_at": 328,
        "payment_preimage": "c91a341fb56284b30cad694d9d115028a9811
        d93d33137a3451295d20ec1801e",
        "description": "description bread",
        "expires_at": 1686054414
    },
    {
        "label": "invoice10..",
        "bolt11": "lntb20n1pj8avx7sp5v4ejgkylx5hqntfmzrmv7cayrywk5
        ra5hzh4lstk2qg4lawgmh4spp5wp77xyevpgetmtey2vgets7fhrle2

        9ph3tps9nkfevvqjdqz3lfqdquv3jhxcmjd9c8g6t0dcsxyun9v9jq

        xqrafscqp2rzjqfcxsh9gr28y6ngphmk90q05ejfydpq89tjjc5rl36

        lfmtcv424hkffd7qqqqdcqqyqqqqlgqqqqqqgq2q9qyysgqgd5kyaya

        qgq3xy8zwyrptwyanacgt68n8nexgrpsnww83lfj6xj30k9xk3y5tjd
       53mtw0ntj0d7kmx3qdd4530yu3fp74hsvpazuv8qpgggksd",
        "payment_hash": "707de3132c0a32bdaf24531195c3c9b8ff9514378ac
        302cec9cb180934028fd2",
        "msatoshi": 2000,
        "amount_msat": "2000msat",
        "status": "paid",
        "pay_index": 20,
        "msatoshi_received": 2000,
        "amount_received_msat": "2000msat",
        "paid_at": 331,
        "payment_preimage": "53be09690097fd023042dd6379dd8029f3eb851
        38099d6fecb5901f623bff2fd",
        "description": "description bread",
        "expires_at": 1686054414
    },
    {
        "label": "invoice11..",
        "bolt11": "lntb20n1pj8avx7sp5pdtd9ykpaej2wgnwyqsv2hj
        4hjj2ml4t78rh50yj9xenc8qsun5qpp5djv8v483w5j2x9pew8ud7qv
        zkmct4thhavur4zqjtmqmh8sxrjwsdquv3jhxcmjd9c8g6t0dcsxyun
        9v9jqxqrafscqp2rzjqfcxsh9gr28y6ngphmk90q05ejfydpq89tjjc
        5rl36lfmtcv424hkffd7qqqqdcqqyqqqqlgqqqqqqgq2q9qyysgqhvu
        xm5juuzz65ppdvsz8yecegj0m6xk3et7yg09h5la49gvquadsyy7g0q
        w8rqqr2l50tt4s8j4zd4xals3tegfk6tur4rx8ystl3mcpktftsp",
        "payment_hash": "6c987654f17524a3143971f8df0182b6f0baaef
        7eb383a88125ec1bb9e061c9d",
        "msatoshi": 2000,
        "amount_msat": "2000msat",
        "status": "paid",
        "pay_index": 21,
        "msatoshi_received": 2000,
        "amount_received_msat": "2000msat",
        "paid_at": 335,
        "payment_preimage": "21f8932d8ee9febac2ce8ae383d4979cefe8
        d4206c61daa5d8119f0321a2eff8",
        "description": "description bread",
        "expires_at": 1686054414
    },
    {
        "label": "invoice12..",
        "bolt11": "lntb20n1pj8avx7sp5mscsvyej780xd56nuj540fxxszg8c3
        vad9rxtpqdny58z7ud4flspp5h2wem9nmq67y6w2r6z7mnsq78nvcaqg
        2kp3g97k3y6qvdjkqastsdquv3jhxcmjd9c8g6t0dcsxyun9v9jqxqra
        fscqp2rzjqfcxsh9gr28y6ngphmk90q05ejfydpq89tjjc5rl36lfmtc
        v424hkffd7qqqqdcqqyqqqqlgqqqqqqgq2q9qyysgq5gtnlxdwqhnaay
        50a4nkzct7fjdquqewnh8nphpthe7w8mht9hy8vgmvwcvn29ut24dan8
        27lqdfe39uzzckc55h7x5lzdfv8qnh2xsq69ga7n",
        "payment_hash": "ba9d9d967b06bc4d3943d0bdb9c01e3cd98e810a
        b06282fad12680c6cac0ec17",
        "msatoshi": 2000,
        "amount_msat": "2000msat",
        "status": "paid",
        "pay_index": 22,
        "msatoshi_received": 2000,
        "amount_received_msat": "2000msat",
        "paid_at": 338,
        "payment_preimage": "813caa2d9b451a52278427b80c30f073ad00d
        347d4ab0004ebecc040bb9c7115",
        "description": "description bread",
        "expires_at": 1686054414
    },
    {
        "label": "invoice13..",
        "bolt11": "lntb20n1pj8avx7sp5paq98vkh3n3a4q0ms8qkv7prlqq5lweh
        k3yhexydnnt7e97nlz4spp5ju0yeh3kjt6e339gd32tayn0awmuqqd
        vz88mvgj7zx0qdtgfkcrsdquv3jhxcmjd9c8g6t0dcsxyun9v9jqxq
        rafscqp2rzjqfcxsh9gr28y6ngphmk90q05ejfydpq89tjjc5rl36l
        fmtcv424hkffd7qqqqdcqqyqqqqlgqqqqqqgq2q9qyysgqa6t7zz40
        nldkwmzjng2tnddj62ga3zngfpfyrm8vnyl80ce40ph3m5a9twj5m37
        revapxse6t3acyqfzayujguuz8k4k84q29sxya8qql9p4el",
        "payment_hash": "971e4cde3692f598c4a86c54be926febb7c001ac
        11cfb6225e119e06ad09b607",
        "msatoshi": 2000,
        "amount_msat": "2000msat",
        "status": "paid",
        "pay_index": 23,
        "msatoshi_received": 2000,
        "amount_received_msat": "2000msat",
        "paid_at": 341,
        "payment_preimage": "b011ee56cfdb9866ae75e9e2dc5d8cf4cff1d2b1
        0150538ca2ebd0c691fda070",
        "description": "description bread",
        "expires_at": 1686054414
    },
    {
        "label": "invoice14..",
        "bolt11": "lntb20n1pj8avx7sp5zvax9a62kucmwjw88czva0ceqxrmk0g
        0n3v3q9lwwxf27sl792wqpp52yj0p92u5qglcpz0vh6yrhl5x4e0pge
        x3k6vu4tu776d38vh06jsdquv3jhxcmjd9c8g6t0dcsxyun9v9jqxqr
        afscqp2rzjqfcxsh9gr28y6ngphmk90q05ejfydpq89tjjc5rl36lfm
        tcv424hkffd7qqqqdcqqyqqqqlgqqqqqqgq2q9qyysgqcq0wplapz0a
        gllnr56e3286pr5a9lkf3p29vjha48m57wsa68wk5y645c8u04uam48

        zapk3nw98l0j57ylka99ts8zl4x5hqg7azx2gqh3276j,
        "payment_hash": "5124f0955ca011fc044f65f441dff43572f0a326
        8db4ce557cf7b4d89d977ea5",
        "msatoshi": 2000,
        "amount_msat": "2000msat",
        "status": "paid",
        "pay_index": 24,
        "msatoshi_received": 2000,
            "amount_received_msat": "2000msat",
            "paid_at": 344,
            "payment_preimage": "b4b9de03542cc25518e5e2259afb0a45aef
            d440ba480bd1713f001a1c2e4d3bc",
            "description": "description bread",
            "expires_at": 1686054414
        },
        {
            "label": "invoice15..",
            "bolt11": "lntb20n1pj8avx7sp593mtlhysggpf7makym9hlcdu264j
            6mey5s83kpkctvukke7pcktspp5ygwyyxfzhayz4jhwq7zgq2y7sclwq2
            gs804rzmfj9qzntgy0jvgsdquv3jhxcmjd9c8g6t0dcsxyun9v9jqxqra
            fscqp2rzjqfcxsh9gr28y6ngphmk90q05ejfydpq89tjjc5rl36lfmtc
            v424hkffd7qqqqdcqqyqqqqlgqqqqqqgq2q9qyysgqnjq3y3ddygu5ux
            dftl47z9yr3xqv2z9vzv94y5k46at5m7ky5xxhv6p224xpnzkrang38w
            d36s2fgf8lg4gcutp8vkrs6w6l3q9rj0cpwt7xh9",
            "payment_hash": "221c421922bf482acaee078480289e863ee02910
            3bea316d32280535a08f9311",
            "msatoshi": 2000,
            "amount_msat": "2000msat",
            "status": "paid",
            "pay_index": 25,
            "msatoshi_received": 2000,
            "amount_received_msat": "2000msat",
            "paid_at": 347,
            "payment_preimage": "50972286a0cd180dda519f5098eee2a62a8
            799d1155ebb9dfd02cc99d995e557",
            "description": "description bread",
            "expires_at": 1686054414
        },
        {
            "label": "invoice16..",
            "bolt11": "lntb20n1pj8avxlsp5z7lqudvxnttyamfrmvht9p0qvmm
            xrksaxtyuhhw90vsx9u4fa5qqpp5hxzwadkr9fc0j57a47s2sgzqyl7dqy
            ducx8xm4ntr7k8vwwy0d2sdquv3jhxcmjd9c8g6t0dcsxyun9v9jqxqra
            fscqp2rzjqfcxsh9gr28y6ngphmk90q05ejfydpq89tjjc5rl36lfmt
            cv424hkffd7qqqqdcqqyqqqqlgqqqqqqgq2q9qyysgqhedawjzgt48s
            u559recfd6g2y42amag68hvpp98l98zj0qg6l2y8mendp8u6vqv8h3d
            rfqrna7guxvanelk4lzrdrvlznm8mytjac5cq03xxd2",
            "payment_hash": "b984eeb6c32a70f953ddafa0a8204027fcd011b
            cc18e6dd66b1fac7639c47b55",
            "msatoshi": 2000,
            "amount_msat": "2000msat",
            "status": "paid",
        "pay_index": 26,
        "msatoshi_received": 2000,
        "amount_received_msat": "2000msat",
        "paid_at": 351,
        "payment_preimage": "fcf91bba50fc49b5c585bd2ca928fea029553d8
        52eb15ff63e15d1a9ff1b2232",
        "description": "description bread",
        "expires_at": 1686054415
    },
    {
        "label": "invoice17..",
        "bolt11": "lntb20n1pj8avxlsp5rvq2asx0832vfqqkvm4q5g6wjc2683m
        vdsw2awg9gsc8pw8ddf2qpp549htt6cqjd9mzhjnn046sm6e3qtek0ld
        az2kvhh7yykl0rhldkjqdquv3jhxcmjd9c8g6t0dcsxyun9v9jqxqraf
        scqp2rzjqfcxsh9gr28y6ngphmk90q05ejfydpq89tjjc5rl36lfmtcv
        424hkffd7qqqqdcqqyqqqqlgqqqqqqgq2q9qyysgqr3fa7hkrmma0a97
        0d2jjzsk4tttdwapg90k3gq6e9x587d2xjm58crdtp4jm7378v4shu2e
        7gadc3h2585u5qal979g8facmp93vpzgpku8ssu",
        "payment_hash": "a96eb5eb00934bb15e539beba86f5988179b3fe
        de895665efe212df78eff6da4",
        "msatoshi": 2000,
        "amount_msat": "2000msat",
        "status": "paid",
        "pay_index": 27,
        "msatoshi_received": 2000,
        "amount_received_msat": "2000msat",
        "paid_at": 355,
        "payment_preimage": "625448f99e9f750fc6ee9aa80d8f3f1a7d9fc99f
        f30b95952c25d4da7d0dde12",
        "description": "description bread",
        "expires_at": 1686054415
    },
    {
        "label": "invoice18..",
        "bolt11": "lntb20n1pj8avxlsp5qj38uvu555fk35zqffjppkpcs0hved
        xkqcxe9fhdpxx8k0avkxkspp596dmf6uxfh2tgd9l6clnpe59c622mx
        mk2fr04s0wml570u2xg63qdquv3jhxcmjd9c8g6t0dcsxyun9v9jqxq
        rafscqp2rzjqfcxsh9gr28y6ngphmk90q05ejfydpq89tjjc5rl36lf
        mtcv424hkffd7qqqqdcqqyqqqqlgqqqqqqgq2q9qyysgq7tffw9j837
        wedad5tps77329mjr843ua0vwthlgpesh5l8z25js98xx8mwpkkdc8d
        crjxh6ueezuyzq8rrge02d78633aachmptvnzqqdagx0z",
        "payment_hash": "2e9bb4eb864dd4b434bfd63f30e685c694ad9b765
        246fac1eedfe9e7f14646a2",
        "msatoshi": 2000,
            "amount_msat": "2000msat",
            "status": "paid",
            "pay_index": 28,
            "msatoshi_received": 2000,
            "amount_received_msat": "2000msat",
            "paid_at": 358,
            "payment_preimage": "72e6ddc9e0cdb7b38a79d15d7aa3015bc1
            3c1ef863b84da08f88163ac2c17458",
            "description": "description bread",
            "expires_at": 1686054415
        },
        {
            "label": "invoice19..",
            "bolt11": "lntb20n1pj8avxlsp5mahg8hgce0pv0d7sy658elhx6ew
            mlaqjjcxkvf6eyr8p037sr8rqpp5l8lsptna2xjzakqg8z5p7p9ws908e
            255lf7ge5x4ls2k0rpd6jtqdquv3jhxcmjd9c8g6t0dcsxyun9v9jqxq
            rafscqp2rzjqfcxsh9gr28y6ngphmk90q05ejfydpq89tjjc5rl36lf
            mtcv424hkffd7qqqqdcqqyqqqqlgqqqqqqgq2q9qyysgqw69pr7p5gv
            msz7nsvf5eymaya2tn9l5j92w6l4k0dhp4cfdelvxyg68auyks2cevc
            jww50q23va5zn456d7kezrl4d48tvvyenzl5gqpu7nudr",
            "payment_hash": "f9ff00ae7d51a42ed80838a81f04ae815e7caa9
            4fa7c8cd0d5fc15678c2dd496",
            "msatoshi": 2000,
            "amount_msat": "2000msat",
            "status": "paid",
            "pay_index": 29,
            "msatoshi_received": 2000,
            "amount_received_msat": "2000msat",
            "paid_at": 362,
            "payment_preimage": "106d843e1cbb4602c405f50a362a523e2174814
            98599a3509351acb89f4bceb9",
            "description": "description bread",
            "expires_at": 1686054415
        },
        {
            "label": "invoice20..",
            "bolt11": "lntb20n1pj8avxlsp5f7kqur55zx7g88ksyjlagpe9eaj
            mhzufzntuxg3ejzluv8z9l7lqpp5tzwy7yrmj2yu7a58kn95ecz2zlkc
            u34etauf0xh9lkweul0jm58qdquv3jhxcmjd9c8g6t0dcsxyun9v9
            jqxqrafscqp2rzjqfcxsh9gr28y6ngphmk90q05ejfydpq89tjjc5r
            l36lfmtcv424hkffd7qqqqdcqqyqqqqlgqqqqqqgq2q9qyysgqgkq0
            ectas5q5nmxpwgwgkh62un64uy6q5vx9u7xmf7g2a27t06ts29cewe
            2sunezr2ju6qfa8vstvw6rg23cmgrlwyc384kn4vxjt4gq50n8v8",
            "payment_hash": "589c4f107b9289cf7687b4cb4ce04a17ed8e46b
        95f78979ae5fd9d9e7df2dd0e",
        "msatoshi": 2000,
        "amount_msat": "2000msat",
        "status": "paid",
        "pay_index": 30,
        "msatoshi_received": 2000,
        "amount_received_msat": "2000msat",
        "paid_at": 365,
        "payment_preimage": "2713eb80f46ff867d1df8ace3b3c468995f9c45
        fb33e311dfb0aeaf91aec1af9",
        "description": "description bread",
        "expires_at": 1686054415
    },
    {
        "label": "invoice21..",
        "bolt11": "lntb20n1pj8avxlsp5pyar4ys88stwjuhtpeu2fl7vj55an5
        ur3swackjeltxpptpavycqpp5zhcncmrlvhmhv7a7h2s4rvunrc0py
        lgrr7ml48xvfd5jku0y4qnqdquv3jhxcmjd9c8g6t0dcsxyun9v9jq
        xqrafscqp2rzjqfcxsh9gr28y6ngphmk90q05ejfydpq89tjjc5rl3
        6lfmtcv424hkffd7qqqqdcqqyqqqqlgqqqqqqgq2q9qyysgqykfc44
        y8frtz303mtn7d9s4ftyxzemfwxx9dfr24l5ueq88plkz9cxspl9xe
        3s0jrg4kppxrcyfzqne9kkte3ela275n9gqgys9lf8gqlkx5df",
        "payment_hash": "15f13c6c7f65f7767bbebaa151b3931e1e127d0
        31fb7fa9ccc4b692b71e4a826",
        "msatoshi": 2000,
        "amount_msat": "2000msat",
        "status": "paid",
        "pay_index": 31,
        "msatoshi_received": 2000,
        "amount_received_msat": "2000msat",
        "paid_at": 368,
        "payment_preimage": "1994f74b38c822ae5cb58ed6e56445aa9f13185
        4c140f75da06ce46e32e5d004",
        "description": "description bread",
        "expires_at": 1686054415
    },
    {
        "label": "invoice22..",
        "bolt11": "lntb20n1pj8avxlsp5k3yuau253hyqr5nx80p0udkvjt23eq7
        5s3f33y09dkh5m5hghlespp5z9xmge0ayvd82hpzlc5wselz75qe28
        ae3y328x602lmq79g7a0kqdquv3jhxcmjd9c8g6t0dcsxyun9v9jqx
        qrafscqp2rzjqfcxsh9gr28y6ngphmk90q05ejfydpq89tjjc5rl36
        lfmtcv424hkffd7qqqqdcqqyqqqqlgqqqqqqgq2q9qyysgql94w8yh
        20v2gwz4u7eud7kc5rdl6tgks79rw9z6tj63k2thtvmd326fm5k2pu
        54767htuqakcf4xjwkypmxt9g4adpkyzrard9p970gpmjrg0d",
        "payment_hash": "114db465fd231a755c22fe28e867e2f501951fb
        98922a39b4f57f60f151eebec",
        "msatoshi": 2000,
        "amount_msat": "2000msat",
        "status": "paid",
        "pay_index": 32,
        "msatoshi_received": 2000,
        "amount_received_msat": "2000msat",
        "paid_at": 372,
        "payment_preimage": "d2160d6a96ee4daf7419e0cf26c5c053f92
        501ff2a6756ea8fb0f5c04439dca8",
        "description": "description bread",
        "expires_at": 1686054415
    },
    {
        "label": "invoice23..",
        "bolt11": "lntb20n1pj8avxlsp5e0xqc9nnd5y95kwvljuky0hd99n
        deemygw46me59jv0v9u2r3teqpp57ftcnex008zcesaak4zxu4v73w8dp
        lrxqyzvh6xugmfh2hg8txjsdquv3jhxcmjd9c8g6t0dcsxyun9v9jq
        xqrafscqp2rzjqfcxsh9gr28y6ngphmk90q05ejfydpq89tjjc5rl3
        6lfmtcv424hkffd7qqqqdcqqyqqqqlgqqqqqqgq2q9qyysgq0gvl9m
        vmsltv9pqjyysn929ru65xztkzw7sm5kz4y8usreawkgk46ee7nyxl
        py3h4fg8yav505ftk2x2yqa94fu7zmcl6lygjjc5nqsp3vyu02",
        "payment_hash": "f25789e4cf79c58cc3bdb5446e559e8b8ed0fc66
        0104cbe8dc46d3755d0759a5",
        "msatoshi": 2000,
        "amount_msat": "2000msat",
        "status": "paid",
        "pay_index": 33,
        "msatoshi_received": 2000,
        "amount_received_msat": "2000msat",
        "paid_at": 375,
        "payment_preimage": "32c1a006fca6111f6967f36229b322261e3
        869c0478ebfc532a35ec300ca98ef",
        "description": "description bread",
        "expires_at": 1686054415
    },
    {
        "label": "invoice24..",
        "bolt11": "lntb20n1pj8avxlsp56rsq4n543jwqy3cxteh7uf0agv
        x2c3n499gm3g9yr90x3qe58f8spp5dqrma3zs49tt6nz2mq7fp08xns
        a72569rj42lyaxnvrzpanx96aqdquv3jhxcmjd9c8g6t0dcsxyun9v9
        jqxqrafscqp2rzjqfcxsh9gr28y6ngphmk90q05ejfydpq89tjjc5r
        l36lfmtcv424hkffd7qqqqdcqqyqqqqlgqqqqqqgq2q9qyysgq2mat
        786vfjf9sxcx6tk826rq5wz89td8h0dgvf8mxyaud9lav3vswrmlzn
        klf32wlertnjlmk2n8z424s9n6s0dzyzxj8q0lw9yuh7qpws20cz",
        "payment_hash": "6807bec450a956bd4c4ad83c90bce69c3be5534
        51caaaf93a69b0620f6662eba",
        "msatoshi": 2000,
        "amount_msat": "2000msat",
        "status": "paid",
        "pay_index": 34,
        "msatoshi_received": 2000,
        "amount_received_msat": "2000msat",
        "paid_at": 379,
        "payment_preimage": "b18f02fce6a8f787c6d04064aa887beeab1755c
        f41f248bcbe5b7d4873d9ebab",
        "description": "description bread",
        "expires_at": 1686054415
    },
    {
        "label": "invoice25..",
        "bolt11": "lntb20n1pj8av8qsp5keh76y3msmt4scjjksx52uj8xct5h2f
        76602w4agrwq6jhhgg5kspp543unqqf6l6f6d97cm3zl7rv3plg4pem
        cs07clmsjr9mdlr3t4azqdquv3jhxcmjd9c8g6t0dcsxyun9v9jqxqr
        afscqp2rzjqfcxsh9gr28y6ngphmk90q05ejfydpq89tjjc5rl36lfm
        tcv424hkffd7qqqqdcqqyqqqqlgqqqqqqgq2q9qyysgqvx3f0rlkmjam
        4n6j7djp08pq5k5qcrg9hx8r4a02tgwwpe4svp58y58zrvkx34u9n2dy
        7p5f2pnkenp6rdhqtpndvlq2yc7h4d48ryqp07xenn",
        "payment_hash": "ac7930013afe93a697d8dc45ff0d910fd150e7
        7883fd8fee121976df8e2baf44",
        "msatoshi": 2000,
        "amount_msat": "2000msat",
        "status": "paid",
        "pay_index": 35,
        "msatoshi_received": 2000,
        "amount_received_msat": "2000msat",
        "paid_at": 382,
        "payment_preimage": "f65dd138d0d577e7d5f6b8066242b0ebb53e72b
        813bd1baed05f2c334a038e80",
        "description": "description bread",
        "expires_at": 1686054416
    },
    {
        "label": "invoice26..",
        "bolt11": "lntb20n1pj8av8qsp5rxg7cfsrttua9muzr8pusmwd8fhp3r5
        j82h4lujeqzm77l5krgqspp5xwz4dlhx39jmjefuw6zj2k67dffsyg9
```

```
      qqcw9mf974ng3lw956m5qdquv3jhxcmjd9c8g6t0dcsxyun9v9jqxqr
      afscqp2rzjqfcxsh9gr28y6ngphmk90q05ejfydpq89tjjc5rl36lfm
      tcv424hkffd7qqqqdcqqyqqqqlgqqqqqqgq2q9qyysgq8mhru326zre
      8y9rd3lwerzgyhp8mq3cprkgms0h0frw8p5w2dvp9ezlzcf9fdcl4u9
      uak2jtju35dy4ksvmcmrjs2kgcdptf4nxyz8cpqqc5ad",
      "payment_hash": "338556fee68965b9653c7685255b5e6a530220a00
      61c5da4beacd11fb8b4d6e8",
      "msatoshi": 2000,
      "amount_msat": "2000msat",
      "status": "paid",
      "pay_index": 36,
      "msatoshi_received": 2000,
      "amount_received_msat": "2000msat",
      "paid_at": 386,
      "payment_preimage": "9f82815920d56c629a5c0f6d0f30d9a7dd58c6
      80187d070a1e5af41e52ea56ed",
      "description": "description bread",
      "expires_at": 1686054416
    },
    {
      "label": "invoice27..",
      "bolt11": "lntb20n1pj8av8qsp56yzky49p4cfj6dcp9pc4f6thp7pmxk
      9c38ye9q5n4nanl9zt6gfspp5gmu4rnexsjhg6w4enhl4kfwjca69
      02jfwg6v37vxczl4pw3r6assdquv3jhxcmjd9c8g6t0dcsxyun9v9
      jqxqrafscqp2rzjqfcxsh9gr28y6ngphmk90q05ejfydpq89tjjc5
      rl36lfmtcv424hkffd7qqqqdcqqyqqqqlgqqqqqqgq2q9qyysgqll
      pxy58salgv8z4njdffn3k34jgf9cpnf6q3lum6hj6r7uyt6hwrsdw
      3srpmm580yec0xpvsa7xw9wny03qrefe0qkne2u0mac99vjqpr75lyy",
      "payment_hash": "46f951cf2684ae8d3ab99dff5b25d2c77457a
      a497234c8f986c0bf50ba23d761",
      "msatoshi": 2000,
      "amount_msat": "2000msat",
      "status": "paid",
      "pay_index": 37,
      "msatoshi_received": 2000,
      "amount_received_msat": "2000msat",
      "paid_at": 389,
      "payment_preimage": "b25325d386d2aa202b81339e3529c07b11671d
      b3d8f684feb4c39f4f488fd60a",
      "description": "description bread",
      "expires_at": 1686054416
    },
    {
      "label": "invoice28..",
        "bolt11": "lntb20n1pj8av8qsp5j5dp4duxcz2h5k48r0vwtqjzvgzd5nk
        r4nugk34y6s8nnh26j0dqpp5n3qve2frxgz48zx4t5g7h6mx6j
        sspqqgzse8hql5d0e3y9ewrhasdquv3jhxcmjd9c8g6t0dcsxy
        un9v9jqxqrafscqp2rzjqfcxsh9gr28y6ngphmk90q05ejfydp
        q89tjjc5rl36lfmtcv424hkffd7qqqqdcqqyqqqqlgqqqqqqgq
        2q9qyysgqlpx25um5lgc46ygkuac9gzj6kpxva6tuattpcry5k
        39uw9vpjp5q02vuacamdacvyyqmzpjurk9mg34kqjgk0hjka2wctlzwq
        6327tsq0vpz84",
        "payment_hash": "9c40cca92332055388d55d11ebeb66d4a1008
        00814327b83f46bf312172e1dfb",
        "msatoshi": 2000,
        "amount_msat": "2000msat",
        "status": "paid",
        "pay_index": 38,
        "msatoshi_received": 2000,
        "amount_received_msat": "2000msat",
        "paid_at": 392,
        "payment_preimage": "1e704d1c00a9ab1377fae3c6855fad14bafd8
        007014c5c18ca1ba04c498f231d",
        "description": "description bread",
        "expires_at": 1686054416
    },
    {
        "label": "invoice29..",
        "bolt11": "lntb20n1pj8av8qsp5an40r8q2n3aph0g0q9ptlhcehy2tk4n
        yz95tv7w27ruxrzueqt0spp5eycke6qf9yp0xf332j794at78d2s4jgd7d
        76fmlpkx5l8gtw5ygqduv3jhxcmjd9c8g6t0dcsxyun9v9jqxqrafscqp
        2rzjqfcxsh9gr28y6ngphmk90q05ejfydpq89tjjc5rl36lfmtcv424hkf
        fd7qqqqdcqqyqqqqlgqqqqqqgq2q9qyysgquln22a70x8e96gajert2cvd
        vwhezzk8ddl7jtu004mh5mjnnnh78ph7utv3tyhkputlsgvzxyvdkp2cu9
        juzr968u5j24lfhejmp0lcpv8vrkn",
        "payment_hash": "c9316ce8092902f3263154bc5af57e3b550ac
        90df37da4efe1b1a9f3a16ea110",
        "msatoshi": 2000,
        "amount_msat": "2000msat",
        "status": "paid",
        "pay_index": 39,
        "msatoshi_received": 2000,
        "amount_received_msat": "2000msat",
        "paid_at": 396,
        "payment_preimage": "81ad8c73eef7a1f99de87801acd282cbc085ba
        69bacb314e19c4500920c1b8d9",
        "description": "description bread",
        "expires_at": 1686054416
    },
    {
      "label": "invoice30..",
      "bolt11": "lntb20n1pj8av8qsp5n4k05g90m96lz69jc5t6ylpnlp9se
      v3v7qfnc0dhgke5d3uunmqspp5g6xp5v6a7qrdx04rk2w7j0k969dxflerk
      svwhrmxufq6dll3pkdqdquv3jhxcmjd9c8g6t0dcsxyun9v9jqxqrafsc
      qp2rzjqfcxsh9gr28y6ngphmk90q05ejfydpq89tjjc5rl36lfmtcv424
      hkffd7qqqqdcqqyqqqqlgqqqqqqgq2q9qyysgqczy75czls96ffpxx3rr
      9patpd506l6q82569dhmx0hja3eerfc8jzl24249tu240g8vktp8gufn0
      vur4htwjaa7zv0y4cr3rx7t4amgpre3avx",
      "payment_hash": "468c1a335df006d33ea3b29de93ec5d15a64f
      f23b418eb8f66e241a6fff10d9a",
      "msatoshi": 2000,
      "amount_msat": "2000msat",
      "status": "paid",
      "pay_index": 40,
      "msatoshi_received": 2000,
      "amount_received_msat": "2000msat",
      "paid_at": 399,
      "payment_preimage": "ccee39e77f17e13d76fbebd955f45f5919289f
      65c2e31b69cade7ba93223371f",
      "description": "description bread",
      "expires_at": 1686054416
    },
    {
      "label": "invoice31..",
      "bolt11": "lntb20n1pj8av8qsp5gjt8n0u99jvylfpfugsn7d8re4se
      zw8a6c6pe8vw04d594lv2ljqpp5ngtv78lvh8ezzv0tytxrnz3kcvpal2
      ye5446vah0xhfzg2hwr0ksdquv3jhxcmjd9c8g6t0dcsxyun9v9jqxqraf
      scqp2rzjqfcxsh9gr28y6ngphmk90q05ejfydpq89tjjc5rl36lfmtcv
      424hkffd7qqqqdcqqyqqqqlgqqqqqqgq2q9qyysgqg68d2hu9lnsp0ut
      m0dymmr94ycwys0tfpujcflxe7l3z3am9rcg3j26ygnext33mukrvv3m
      tqacmpak6rk6mqhja4psgd6uccu9eglgpcu9z4h",
      "payment_hash": "9a16cf1fecb9f22131eb22cc398a36c303dfa
      899a56ba676ef35d2242aee1bed",
      "msatoshi": 2000,
      "amount_msat": "2000msat",
      "status": "paid",
      "pay_index": 41,
      "msatoshi_received": 2000,
      "amount_received_msat": "2000msat",
      "paid_at": 402,
      "payment_preimage": "7941a6bcf038ce924c640055bf152acecaa6b
      c1fee9edec37aa9d22b7e707be9",
        "description": "description bread",
        "expires_at": 1686054416
    },
    {
        "label": "invoice32..",
        "bolt11": "lntb20n1pj8av8psp5uz7j6f332yy3kv676qtalgs4e2wzkn
        zff0tgg9r65scp5zn3afsspp54frx9jccn0yq6m84wsgavmqkks56sz
        nndplcjr29d6np5w6qg0esdquv3jhxcmjd9c8g6t0dcsxyun9v9jqxq
        rafscqp2rzjqfcxsh9gr28y6ngphmk90q05ejfydpq89tjjc5rl36lf
        mtcv424hkffd7qqqqdcqqyqqqqlgqqqqqqgq2q9qyysgq3gpv53nxux
        xgce9x2ury5ahnypc96q3vq90hd4ath9x5ap4tva4922fnprmrdphvd
        jpcrc0f9ze3zn8q7nuumcc7zwt7u0525kxzufgpezj4st",
        "payment_hash": "aa4662cb189bc80d6cf57411d66c16b429a80a
        73687f890d456ea61a3b4043f3",
        "msatoshi": 2000,
        "amount_msat": "2000msat",
        "status": "paid",
        "pay_index": 42,
        "msatoshi_received": 2000,
        "amount_received_msat": "2000msat",
        "paid_at": 406,
        "payment_preimage": "c655dfb3f89068e27b2a818c166be537f0c5ea6
        601c876a9e2bfca69519b4269",
        "description": "description bread",
        "expires_at": 1686054417
    },
    {
        "label": "invoice33..",
        "bolt11": "lntb20n1pj8av8psp5lm2nkmy4w34a8f0y2e2ggmslskwarg
        zhr0qkptda6rjq4uzgmmvspp53r27yw99s04gtnfhtn7cv5nczvrahne
        y5277ymef9ftpvhhes3qqdquv3jhxcmjd9c8g6t0dcsxyun9v9jqxqra
        fscqp2rzjqfcxsh9gr28y6ngphmk90q05ejfydpq89tjjc5rl36lfmtc
        v424hkffd7qqqqdcqqyqqqqlgqqqqqqgq2q9qyysgqpeyq7udfllkvzd
        ryxa6j59mn62gr6kmzuut85xrysmlfe3tjpw890anyn8wpwlm6drtnxa
        r37gc8u2664hqtucjy7x7uj7cjmy99jqsqdqxaeq",
        "payment_hash": "88d5e238a583ea85cd375cfd8652781307dbcf
        24a2bde26f292a56165ef98440",
        "msatoshi": 2000,
        "amount_msat": "2000msat",
        "status": "paid",
        "pay_index": 43,
        "msatoshi_received": 2000,
        "amount_received_msat": "2000msat",
        "paid_at": 409,
        "payment_preimage": "dae467b509839eb1d7587c57f45a24330a9392
        5e227766eca6e254dcf50895ae",
        "description": "description bread",
        "expires_at": 1686054417
    },
    {
        "label": "invoice34..",
        "bolt11": "lntb20n1pj8av8psp57en7yk6wxzjdkft5ugtpwaknm74qm7s
        ctn0qvgxepl0gqhh5x0qspp5wvkwd2zplg5jhlhrtlsddxtz3pvd3ehy
        5lwxjj06s032dlrtxzxqdquv3jhxcmjd9c8g6t0dcsxyun9v9jqxqraf
        scqp2rzjqfcxsh9gr28y6ngphmk90q05ejfydpq89tjjc5rl36lfmtcv
        424hkffd7qqqqdcqqyqqqqlgqqqqqqgq2q9qyysgqcjx9a4k8r3fwh3q
        ykn89rc3rsa9vp2dtrd7hqxggapfuhthvquzxlvwgztv022qdc5l3ye3
        mnvmmd005ze8nyfgtc0wxlkvgacpx5psqnk4mpw",
        "payment_hash": "732ce6a841fa292bfee35fe0d699628858d8e6
        e4a7dc6949fa83e2a6fc6b308c",
        "msatoshi": 2000,
        "amount_msat": "2000msat",
        "status": "paid",
        "pay_index": 44,
        "msatoshi_received": 2000,
        "amount_received_msat": "2000msat",
        "paid_at": 413,
        "payment_preimage": "4faa28850bcd4832038fa5b2b577dcc468af41b
        4ac00ee612b6e7ed601712fcf",
        "description": "description bread",
        "expires_at": 1686054417
    },
    {
        "label": "invoice35..",
        "bolt11": "lntb20n1pj8av8psp5k9d73zw0937y6r6funmlstzslrvtk
        qpp5clshgh6cpjnrh79asmqpp5rdn06vw9jlxq8dhyejlacqf6uja5f9hj3
        q4zzlzaf68zdu3ak2nqdquv3jhxcmjd9c8g6t0dcsxyun9v9jqxqrafscqp2
        rzjqfcxsh9gr28y6ngphmk90q05ejfydpq89tjjc5rl36lfmtcv424hkff
        d7qqqqdcqqyqqqqlgqqqqqqgq2q9qyysgqzhke44997desdyzr5kkp24n7
        xzgawzd696z4wsaedm4jzc65ejrqu6gc7mmgud8xl2lwt835vhw7syeand
        07kj7vw7a696vkgedca6qq0w99nk",
        "payment_hash": "1b66fd31c597cc03b6e4ccbfdc013ae4bb44
        96f2882a217c5d4e8e26f23db2a6",
        "msatoshi": 2000,
        "amount_msat": "2000msat",
        "status": "paid",
        "pay_index": 45,
        "msatoshi_received": 2000,
        "amount_received_msat": "2000msat",
        "paid_at": 416,
        "payment_preimage": "d78ed5f531b76dc039593a535c7e6a2050668
        cacd3b2f254bb2ea9b9e8fced6c",
        "description": "description bread",
        "expires_at": 1686054417
    },
    {
        "label": "invoice36..",
        "bolt11": "lntb20n1pj8av8psp5854sm20yfgkyz6f7jt9w0a2h673lnpls
        27p6cmxw2qkjuzhu2tgqpp56kqaq36x04fnrfcgxxn4nmcqd3dwrjw6
        aud73hgfc0pxw59smayqdquv3jhxcmjd9c8g6t0dcsxyun9v9jqxqra
        fscqp2rzjqfcxsh9gr28y6ngphmk90q05ejfydpq89tjjc5rl36lfmt
        cv424hkffd7qqqqdcqqyqqqqlgqqqqqqgq2q9qyysgqevyerh7madrz
        9lzrh7lwhuxt8glphlvc893n7u805qcy5js0hykpvzfdf02698fsfdy
        el65j20w7t468qn3sxv7l9tceupduvgdn2kcq8r27tw",
        "payment_hash": "d581d047467d5331a70831a759ef006c5ae1c
        9daef1be8dd09c3c26750b0df48",
        "msatoshi": 2000,
        "amount_msat": "2000msat",
        "status": "paid",
        "pay_index": 46,
        "msatoshi_received": 2000,
        "amount_received_msat": "2000msat",
        "paid_at": 419,
        "payment_preimage": "dbb983d2197798884fb4c849d7f1abff9e003
        3be03202ea93ce8605c8438bced",
        "description": "description bread",
        "expires_at": 1686054417
    },
    {
        "label": "invoice37..",
        "bolt11": "lntb20n1pj8av8psp5ry2nkcynhgsynk8070n754aju2drc8
        she09puwx4a74wal0v4nlqpp5zkk68j8taxcuzstwlk47ftv09uhn3n
        rpkln4c9xl77sh86wgz6nqdquv3jhxcmjd9c8g6t0dcsxyun9v9jqxq
        rafscqp2rzjqfcxsh9gr28y6ngphmk90q05ejfydpq89tjjc5rl36lfm
        tcv424hkffd7qqqqdcqqyqqqqlgqqqqqqgq2q9qyysgq92v8cq88p72a
        ltlavutcps87vx5ytca73lzaefpurhcslpc2t28js4gmvrlyrt3lx7hr
        n06aqnlqqp5s9wl7nc39c5vu78yy44gjsegqc9zra0",
        "payment_hash": "15ada3c8ebe9b1c1416efdabe4ad8f2f2f38c
        c61b7e75c14dff7a173e9c816a6",
        "msatoshi": 2000,
        "amount_msat": "2000msat",
        "status": "paid",
      "pay_index": 47,
      "msatoshi_received": 2000,
      "amount_received_msat": "2000msat",
      "paid_at": 423,
      "payment_preimage": "cf403309bb27b90b19e255e05f735f5cf9bc3
      48270f2643380aa2727568b64f4",
      "description": "description bread",
      "expires_at": 1686054417
   },
   {
      "label": "invoice38..",
      "bolt11": "lntb20n1pj8av8psp5muy4vcc4tgnknfteaesvh4gy65f
      7c20240eu0sfkg34eld8mpfkspp5kv8we2t7flshq5wd0rpqat2spc4847
      80f6l58959lxnqa2epyt2sdquv3jhxcmjd9c8g6t0dcsxyun9v9jqxqraf
      scqp2rzjqfcxsh9gr28y6ngphmk90q05ejfydpq89tjjc5rl36lfmtcv
      424hkffd7qqqqdcqqyqqqqlgqqqqqqgq2q9qyysgqgflk69jtavx30ur
      avt8zgweqp4r6g6knps2s9g78gwldhqsslu3rgmdnsz56svvx2mtkcdr
      h6vdaexn62pkdmzwlv6tn0ylthxqwl0gp30llyw",
      "payment_hash": "b30eeca97e4fe17051cd78c20ead500e2a7af
      8ef4ebf439685f9a60eab2122d5",
      "msatoshi": 2000,
      "amount_msat": "2000msat",
      "status": "paid",
      "pay_index": 48,
      "msatoshi_received": 2000,
      "amount_received_msat": "2000msat",
      "paid_at": 426,
      "payment_preimage": "fd0e76278e2b9417fdcf76edf1bcccbb0001
      a6643addf5faf5dd59325024f505",
      "description": "description bread",
      "expires_at": 1686054417
   },
   {
      "label": "invoice39..",
      "bolt11": "lntb20n1pj8av8psp5nd8zaehwpvqg0ja7y7nul06cldhj
      laueme0xjdt8efrcl6l7m4tspp59za7dqllkp8uydmuznyvyxxk6wypmsua
      9hcwh7ctxuc3fag5amusdquv3jhxcmjd9c8g6t0dcsxyun9v9jqxqrafscq
      p2rzjqfcxsh9gr28y6ngphmk90q05ejfydpq89tjjc5rl36lfmtcv424h
      kffd7qqqqdcqqyqqqqlgqqqqqqgq2q9qyysgqmdj000a0fzvvvxq76hjfr
      q25l7r5hwslsjnyctq96f5qja62ptmlhqf45newj3dsytq4xh8320aeq9
      8g3ka2v0xvtvgyu0p8srqn4ncqqk52ulm",
      "payment_hash": "28bbe683ffb04fc2377c14c8c218d6d3881dc3
      9d2df0ebfb0b373114f514eef9",
      "msatoshi": 2000,
            "amount_msat": "2000msat",
            "status": "paid",
            "pay_index": 49,
            "msatoshi_received": 2000,
            "amount_received_msat": "2000msat",
            "paid_at": 430,
            "payment_preimage": "2a8b0eccb76f1cd7f15ae324d66dc7ee09b2b6
            84705fb339ea34589e033a0a85",
            "description": "description bread",
            "expires_at": 1686054417
        },
        {
            "label": "invoice40..",
            "bolt11": "lntb20n1pj8av8psp5wq5486l404m025ff3l9ytsla92ar9ww
            e4cq0zgrguzevaegn9alspp54h57t90nht4nyttnlewfnmf5cyg5xu6u
            prm24wjp4zkgjus5kzssdquv3jhxcmjd9c8g6t0dcsxyun9v9jqxqraf
            scqp2rzjqfcxsh9gr28y6ngphmk90q05ejfydpq89tjjc5rl36lfmtcv
            424hkffd7qqqqdcqqyqqqqlgqqqqqqgq2q9qyysgq753kskcz3d2vv8n
            rpjncemg6x6npm8xnguarm8uhqcvpavq5q6mx79d7cxhpz5jv6auc4t3
            h8gy55gqlsvg9dlwhuwd2e6xljfaernqqfwvwva",
            "payment_hash": "ade9e595f3baeb322d73fe5c99ed34c1114373
            5c08f6aaba41a8ac897214b0a1",
            "msatoshi": 2000,
            "amount_msat": "2000msat",
            "status": "paid",
            "pay_index": 50,
            "msatoshi_received": 2000,
            "amount_received_msat": "2000msat",
            "paid_at": 433,
            "payment_preimage": "055ba292e8f877d1247efc83f531fcf302faed
            806e54bdd6dfeda5339bf48c8e",
            "description": "description bread",
            "expires_at": 1686054417
        }
    ]
```

Listing A.10 Lighnting Experiment 02 - Channel Node 1 after receiving invoices payments

```
"channels": [
  {
      "peer_id":
"0270685ca81a8e4d4d01beec5781f4cc924684072ae52c507f8ebe9daf0caaab7b",
      "connected": true,
      "state": "CHANNELD_NORMAL",
      "short_channel_id": "2436638x58x1",
      "channel_sat": 3780,
      "our_amount_msat": "3780000msat",
      "channel_total_sat": 100000,
      "amount_msat": "100000000msat",
      "funding_txid":
"c07a7c0c0e1f87ed643a63a0b3bb39198e00df75f60ef8cde849bbb35b153a28",
      "funding_output": 1
   }
]
```

Listing A.11 Node 1 invoices for the experiment 02

```
{
 "label": "invoice02...",
 "bolt11": "lntb200n1pjgqf47sp59u5ghx27ajtpcy8vdvusz6j2rygak8rwd6c
 hxnl4w5fhcsy26c2qpp575h8vn80x9zjnj3gf6shxq7e7y73hqvycl2e5q05mklx0cd
 lvhlqdquv3jhxcmjd9c8g6t0dcsxyun9v9jqxqrafscqp2rzjqfcxsh9gr28y6ng
 phmk90q05ejfydpq89tjjc5rl36lfmtcv424hkffwrcqqqwsqqyqqqlgqqqqqqg
 q2q9qyysgqfyhghph267re9066kee7rlqrr8uudyft5nvv5h8wn6d4v8ej8395u9
 zuy4ns32qpx95j0ymfqckr7x0djfzsmgq5k3ay2qzae4fqgtcpn8up44",
 "payment_hash": "f52e764cef314529ca284ea17303d9f13d1b8184c7d59a
 01f4ddbe67e1bf65fe",
 "msatoshi": 20000,
 "amount_msat": "20000msat",
 "status": "paid",
 "pay_index": 51,
 "msatoshi_received": 20000,
 "amount_received_msat": "20000msat",
 "paid_at": 1686121042,
 "payment_preimage": "34a05997e70b6ed8d25b57cb68dae3d087152f7884552
 f106dbc5f0bc0ea0e5c",
 "description": "description bread",
 "expires_at": 1686150126
},
{
 "label": "invoice03...",
 "bolt11": "lntb200n1pjgqf47sp59rqfy24a5tm75twd5ns2qf9dp93lktjrsv6
 6h22kcg5d8jt8zvpspp5zf7e0kwzsqxv98zsc5scjq2njtd47vdgu97k39mze0jxuj
 y6kh7sdquv3jhxcmjd9c8g6t0dcsxyun9v9jqxqrafscqp2rzjqfcxsh9gr28y6ngp
 hmk90q05ejfydpq89tjjc5rl36lfmtcv424hkffwrcqqqwsqqyqqqlgqqqqqqgq
 2q9qyysgqjztwkj2ju0lr40yg0qyhhqwjjscewperszwvrhvhfhmrp3v53vxyhca
 gq8c6kyjqa0uj35tze7ujx9rpkrjnhez05l3jdy4nnu38cwqqr54ujw",
 "payment_hash": "127d97d9c2800cc29c50c52189015392db5f31a8e17d6
 89762cbe46e489ab5fd",
 "msatoshi": 20000,
 "amount_msat": "20000msat",
 "status": "paid",
 "pay_index": 52,
 "msatoshi_received": 20000,
 "amount_received_msat": "20000msat",
 "paid_at": 1686121085,
 "payment_preimage": "16ea1c70df51426ab0f6dc0c2b8047758824d9498b7f6
 eaffef7cabf6a362c02",
 "description": "description bread",
 "expires_at": 1686150126
},
{
 "label": "invoice04...",
 "bolt11": "lntb200n1pjgqf4lsp5ztcld0r8ue34p7st7j0v850nh0m25xu8sy0234
 ecv3trxwkw5asspp5pzmlanf5pycvgy230ghhvz207237k4rl2378uuxja6u393
 ghat0sdquv3jhxcmjd9c8g6t0dcsxyun9v9jqxqrafscqp2rzjqfcxsh9gr28y6
 ngphmk90q05ejfydpq89tjjc5rl36lfmtcv424hkffwrcqqqwsqqyqqqqlgqqqq
 qqgq2q9qyysgq8yp7hvrzj42qk2dxny8lc5w62754fhtf9eg2d746pvllcvhu5r
 69e6pr4rya3dpuv5cwh8njmdsrgtcnxltlaydrze4qqhsylt4pekgqwhuctp",
 "payment_hash": "08b7fecd340930c411517a2f76094ff2a3eb547f547
 c7e70d2eeb912c517eadf",
 "msatoshi": 20000,
 "amount_msat": "20000msat",
 "status": "paid",
 "pay_index": 53,
 "msatoshi_received": 20000,
 "amount_received_msat": "20000msat",
 "paid_at": 1686121088,
 "payment_preimage": "53dff9e1a6708356aeca5e80c79864ed248f9e7b42c
 543f7b95555062a2adaa2",
 "description": "description bread",
 "expires_at": 1686150127
},
{
 "label": "invoice05...",
 "bolt11": "lntb200n1pjgqf4lsp5qw0dgh8v4asj493xhf0s30e9g9tz7uzntqdvhw
 cqtvuv4552syespp5pp6dhvzekz32wc2vaf487wnsqguq8fs5r9nc0mg2xfu32e9
 lwchqdquv3jhxcmjd9c8g6t0dcsxyun9v9jqxqrafscqp2rzjqfcxsh9gr28y6ng
 phmk90q05ejfydpq89tjjc5rl36lfmtcv424hkffwrcqqqwsqqyqqqqlgqqqqqg
 q2q9qyysgq47edxg3au58uyjx7765c25wgs66fg8uqly638vjjm2s9flvdqrgstv
 5tug4cycmsrgywwmxqyjq46azkx73s5546dh2648w995ncqkgqkgc2g5",
 "payment_hash": "0874dbb059b0a2a7614cea6a7f3a70023803a61419678
 7ed0a32791564bf762e",
 "msatoshi": 20000,
 "amount_msat": "20000msat",
 "status": "paid",
 "pay_index": 54,
 "msatoshi_received": 20000,
 "amount_received_msat": "20000msat",
 "paid_at": 1686121092,
 "payment_preimage": "812ea012310475c005991ef1b272bb64efbecfd469fa1
 69d0d577bced0e3eb72",
 "description": "description bread",
 "expires_at": 1686150127
},
{
 "label": "invoice06...",
 "bolt11": "lntb200n1pjgqf4lsp54gvwys85sdvrhndp2cjcaq4rpvdmm7eqn8n
 893w0ce6c7yqly48spp5deykdghwjxfaf3y0jschpvjtuqt79ftn3sgxgem9ww8mjv
 vf5lfsdquv3jhxcmjd9c8g6t0dcsxyun9v9jqxqrafscqp2rzjqfcxsh9gr28y6ng
 phmk90q05ejfydpq89tjjc5rl36lfmtcv424hkffwrcqqqwsqqyqqqlgqqqqqqg
 q2q9qyysgqxwhd9ysmre07cjlm3p5xy5cuqmxty6afkqvg3s5yamzyvr9waacsjv
 py8cg6kxyg9yw0d7jlqk3ayje3hr6esharg2npfvukaedz4cgq5dh8gj",
 "payment_hash": "6e4966a2ee9193d4c48f943170b24be017e2a5738c10
 646765738fb93189a7d3",
 "msatoshi": 20000,
 "amount_msat": "20000msat",
 "status": "paid",
 "pay_index": 55,
 "msatoshi_received": 20000,
 "amount_received_msat": "20000msat",
 "paid_at": 1686121095,
 "payment_preimage": "813ac2c9c8d7e2f0ed908cd3c9faeaf217f80deed68b
 5e20b2ec2bb31e3328f6",
 "description": "description bread",
 "expires_at": 1686150127
},
{
 "label": "invoice07...",
 "bolt11": "lntb200n1pjgqf4lsp5q3kdcm8d3tkxjlfcjuvkwk70q7tfze9cp68ur
 wj84wv4nj089w3qpp5ant7wdzkq9xjzgfm24y2ff3wzlu2j4u6mqhhzjf2uehs
 qlvgf3cqdquv3jhxcmjd9c8g6t0dcsxyun9v9jqxqrafscqp2rzjqfcxsh9gr2
 8y6ngphmk90q05ejfydpq89tjjc5rl36lfmtcv424hkffwrcqqqwsqqyqqqlg
 qqqqqqgq2q9qyysgqe96ek6pzaxtl7y5p3f4mn6h60889xc7qhmzshfmhazkdf
 8tftlssywvyjgc0pddqg2tad4gnkc595d5rt46pj7wfkkqv5pjy07akfvqqslw2h8",
 "payment_hash": "ecd7e73456014d21213b5548a4a62e17f8a9579ad82f7
 1492ae66f007d884c70",
 "msatoshi": 20000,
 "amount_msat": "20000msat",
 "status": "paid",
 "pay_index": 56,
 "msatoshi_received": 20000,
 "amount_received_msat": "20000msat",
 "paid_at": 1686121099,
 "payment_preimage": "40d5b91bcd60886580baa83ca83f779c39975429686c
 8fcc79799c0ac9674d57",
 "description": "description bread",
 "expires_at": 1686150127
},
{
 "label": "invoice08...",
 "bolt11": "lntb200n1pjgqf4lsp5v08mxm6kwfwmapxs8k90wz9gz0f5vnek03y4a33
 jlzvf5hedrnfspp5ddq3h3w8ldrdf876ya229jgq047s4cpqlmhvze4z35khlyslt
 67qdquv3jhxcmjd9c8g6t0dcsxyun9v9jqxqrafscqp2rzjqfcxsh9gr28y6ngphm
 k90q05ejfydpq89tjjc5rl36lfmtcv424hkffwrcqqqwsqqyqqqqlgqqqqqqgq2q9
 qyysgqlqxacjrtpz5ap0y7h5vznzchn4dm3kejrxy266vt0lfnj6tdnw2jkhw4jmu
 rv0vecnpt3p960jg95tzwqk7kyvhdcs5dhjqq7t7txjsqh7pd3t",
 "payment_hash": "6b411bc5c7fb46d49fda2754a2c9007d7d0ae020feee
 c166a28d2d7f921f5ebc",
 "msatoshi": 20000,
 "amount_msat": "20000msat",
 "status": "paid",
 "pay_index": 57,
 "msatoshi_received": 20000,
 "amount_received_msat": "20000msat",
 "paid_at": 1686121102,
 "payment_preimage": "11bf5cbf969a5f9dec4e59410c7896ba2c2e809a763c
 05b34310f1f076f1ca0f",
 "description": "description bread",
 "expires_at": 1686150127
},
{
 "label": "invoice09...",
 "bolt11": "lntb200n1pjgqf4lsp5v7xpvnq99udjtqrcvkvmlyl7rdn3ldskqwylys
 nhpws2gy45zgespp568a5w9y54cpawnypck8drh7xknh0rnf7ce0lrw7vp8upk2
 7qezjqdquv3jhxcmjd9c8g6t0dcsxyun9v9jqxqrafscqp2rzjqfcxsh9gr28y6
 ngphmk90q05ejfydpq89tjjc5rl36lfmtcv424hkffwrcqqqwsqqyqqqqlgqqqq
 qqgq2q9qyysgqadachhjp5aw0lkj2eg7ge3y95t6zy7sdndxye5rae93ztk93z9
 wswtynutpgtcc8xqhszrf5jkl6c5rhrdypz3vuwc79lcz9llj2rqcp6cjpwx",
 "payment_hash": "d1fb471494ae03d74c81c58ed1dfc6b4eef1cd3ec65
 ff1bbcc09f81b2bc0c8a4",
 "msatoshi": 20000,
 "amount_msat": "20000msat",
 "status": "paid",
 "pay_index": 58,
 "msatoshi_received": 20000,
 "amount_received_msat": "20000msat",
 "paid_at": 1686121105,
 "payment_preimage": "f831749594a3ec55c45921e9cd9cd962442b7f5815f5
 a183a58f9767952c59a5",
 "description": "description bread",
 "expires_at": 1686150127
},
{
 "label": "invoice10...",
 "bolt11": "lntb200n1pjgqf4lsp5tepzd580928t9le2d4aaz8eujvj6dmdez77
 4p4ra942s7a5st9hspp5scjelt3v7mxvznez8kutzfj44tvletkkfm25czcl0xc6k6
 as9svsdquv3jhxcmjd9c8g6t0dcsxyun9v9jqxqrafscqp2rzjqfcxsh9gr28y6ngp
 hmk90q05ejfydpq89tjjc5rl36lfmtcv424hkffwrcqqqwsqqyqqqqlgqqqqqqgq2
 q9qyysgqmaq42rp7h9ac4pwm4nhvhac5llx84e06xcjqrkuc5nl8ckny0czqr5z00
 h2fygl3s29r2q2c9ad8emf946gfnw4yw8ap7q8c5jxtkhqpkzsyse",
 "payment_hash": "86259fae2cf6ccc14f223db8b12655aad9fcaed64e
 d54c0b1f79b1ab6bb02c19",
 "msatoshi": 20000,
 "amount_msat": "20000msat",
 "status": "paid",
 "pay_index": 59,
 "msatoshi_received": 20000,
 "amount_received_msat": "20000msat",
 "paid_at": 1686121109,
 "payment_preimage": "d3b25afd61350da805b5ad025170c4c5d6a41afa13e
 83c66da796dc223327d67",
 "description": "description bread",
 "expires_at": 1686150127
},
{
 "label": "invoice11...",
 "bolt11": "lntb200n1pjgqfkqsp5u0a57u22j6cknln5wp0psmcemusenmmq6p
 t0p3frshu2syddcd7qpp5pd93u4lqz29ax0nws8cmt04sccpmsyzxug7ngmk5c0sr
 j8c6hxqqdquv3jhxcmjd9c8g6t0dcsxyun9v9jqxqrafscqp2rzjqfcxsh9gr28y6
 ngphmk90q05ejfydpq89tjjc5rl36lfmtcv424hkffwrcqqqwsqqyqqqqlgqqqq
 qqgq2q9qyysgqjrshz6gxd9whz76kvf4tz8r7v2zwxmaycd2zt0umg0rg2yuvmx
 xrpx5xydm82xt6hr4ch76gh0arv6zhu86r0rwyqtk27u0ztnlz8agpj5fxvd",
 "payment_hash": "0b4b1e57e0128bd33e6e81f1b5beb0c603b81046e23d346
 ed4c3e0391f1ab980",
 "msatoshi": 20000,
 "amount_msat": "20000msat",
 "status": "paid",
 "pay_index": 60,
 "msatoshi_received": 20000,
 "amount_received_msat": "20000msat",
 "paid_at": 1686121112,
 "payment_preimage": "d806e7caa499a76fdae553e8390866f15281d600e5825ccf
 de9ee6d93da0489d",
 "description": "description bread",
 "expires_at": 1686150128
},
{
 "label": "invoice12...",
 "bolt11": "lntb200n1pjgqfkqsp559f4rlte2el5kzuqn0sq2pj2pm83kfh6ad9wr6qu
 vn7h028cvhkspp5385ar0avlqly84nds7k7u88mr9xtquywm34yw7sahm70qz77wr0
 sdquv3jhxcmjd9c8g6t0dcsxyun9v9jqxqrafscqp2rzjqfcxsh9gr28y6ngphmk90
 q05ejfydpq89tjjc5rl36lfmtcv424hkffwrcqqqwsqqyqqqqlgqqqqqqgq2q9qyys
 gqt0a3d8m7u3kjcyskny6qreacg34j5m2qx6c0zcjp37rgs80zcvw32rw0gp273c34
 nzue09spdn7lpk5y0p9q5eu97xmf8y9mws97cygpxcsnl8",
 "payment_hash": "89e9d1bfacf83e43d66d87adee1cfb194cb0708edc6a
 477a1dbefcf00bde70df",
 "msatoshi": 20000,
 "amount_msat": "20000msat",
 "status": "paid",
 "pay_index": 61,
 "msatoshi_received": 20000,
 "amount_received_msat": "20000msat",
 "paid_at": 1686121116,
 "payment_preimage": "dfb582b416fce7fb9f16da8e6b4bd6c6d1b16c079c5b8
 3abe8b2d9c416b6cb6b",
 "description": "description bread",
 "expires_at": 1686150128
},
{
 "label": "invoice13...",
 "bolt11": "lntb200n1pjgqfkqsp562s3nne3v5fgrakw62yknwl6r6x42fq66f8nr8f
 qe3fqyhw9kuaspp5xglrhdhumdzkkj9mq8qp5zssde0alrn7lu5qs8073ues8u4n
 6wtqdquv3jhxcmjd9c8g6t0dcsxyun9v9jqxqrafscqp2rzjqfcxsh9gr28y6ngp
 hmk90q05ejfydpq89tjjc5rl36lfmtcv424hkffwrcqqqwsqqyqqqqlgqqqqqqgq
 2q9qyysgq45qzd8nqv2ykyxjgqdgjzr559tvlwzvswrxpl4vyvg87t2kyeh2qr5e
 3su2wjtzap424f2kmmr0ayreal8nsv7af0tnsheczfuzc9wcpr5l3et",
 "payment_hash": "323e3bb6fcdb456b48bb01c01a0a106e5fdf8e7eff2
 8081dfe8f3303f2b3d396",
 "msatoshi": 20000,
 "amount_msat": "20000msat",
 "status": "paid",
 "pay_index": 62,
 "msatoshi_received": 20000,
 "amount_received_msat": "20000msat",
 "paid_at": 1686121119,
 "payment_preimage": "79a2ab0b3cd10c88920b68de974bdd24b9ddfe900ba5e
 41c7afb6074ff8cf4c3",
 "description": "description bread",
 "expires_at": 1686150128
},
{
 "label": "invoice14...",
 "bolt11": "lntb200n1pjgqfkqsp50twlmkt8d465r6fcvhl3e8fzu5epgdmsudr
 ywux4r3v7djh4uw5spp57pw8swpy29wr4293l0cm0g9pjwglkpevxx0h27nrcfh5tj
 pf3c7qdquv3jhxcmjd9c8g6t0dcsxyun9v9jqxqrafscqp2rzjqfcxsh9gr28y6
 ngphmk90q05ejfydpq89tjjc5rl36lfmtcv424hkffwrcqqqwsqqyqqqqlgqqqq
 qqgq2q9qyysgqd59x8ggls6qq254fj86jj56tu050chmlestnw0d9ldxqlt536qc
 jhwk4vtr9h7jgsayeahfkeydn97ph2dq6ry968xhuz5xn4jfx6qgpfvrwcr",
 "payment_hash": "f05c783824515c3aa8b1fbf1b7a0a19391fb072c319
 f757a63c26f45c8298e3c",
 "msatoshi": 20000,
 "amount_msat": "20000msat",
 "status": "paid",
 "pay_index": 63,
 "msatoshi_received": 20000,
 "amount_received_msat": "20000msat",
 "paid_at": 1686121122,
 "payment_preimage": "611e089ca2ecdbfd2762bcc76087d7dfdf927d092839
 09ade6b088fae01ac693",
 "description": "description bread",
 "expires_at": 1686150128
},
{
 "label": "invoice15...",
 "bolt11": "lntb200n1pjgqfkqsp56cqccp0yk355l3lqs00f6lyk7zg7adxfm0fg
 wp58p5nmvqujcplqpp5clw6ypz7etw67vg9hy7hkgcmlphg9a3cn644e882nu3ywul59
 ngsdquv3jhxcmjd9c8g6t0dcsxyun9v9jqxqrafscqp2rzjqfcxsh9gr28y6ngphm
 k90q05ejfydpq89tjjc5rl36lfmtcv424hkffwrcqqqwsqqyqqqqlgqqqqqqgq2q9
 qyysgq4p9kap7xcqk8uuq7766jlzwzluqlelh4k36hgsvzdsp46l040pujehx02g0
 wwrl6u008mx0pew4f393yrhqs958l4a6j9q89elwyw0qq25ekyh",
 "payment_hash": "c7dda2045ecaddaf3105b93d7b231bf86e82f6389eab5
 c9cea9f224773f42cd1",
 "msatoshi": 20000,
 "amount_msat": "20000msat",
 "status": "paid",
 "pay_index": 64,
 "msatoshi_received": 20000,
 "amount_received_msat": "20000msat",
 "paid_at": 1686121126,
 "payment_preimage": "e6e7008083e0b53b527573b268ad7396f62d3c19419604
 54b3613ed61a75fb40",
 "description": "description bread",
 "expires_at": 1686150128
},
{
 "label": "invoice16...",
 "bolt11": "lntb200n1pjgqfkqsp5vxes4vrjd928vr5pxjmt822tsx2nfmnhr5x43l30
 ycn08sh9gz4spp5x6jxegpmp0sjjnwec22a59f7p7w040yc65kdqzpk0l2e7px88nv
 qdquv3jhxcmjd9c8g6t0dcsxyun9v9jqxqrafscqp2rzjqfcxsh9gr28y6ngphmk90
 q05ejfydpq89tjjc5rl36lfmtcv424hkffwrcqqqwsqqyqqqqlgqqqqqqgq2q9qyys
 gqlyuxz6eaec5wjhan2mkxyf53dwtt9kfdgpce5xhmyzyp4fjgyykqeas5ez9m2jsz
 88j4hfqtf0dntyeujgr2dgr74a2mhpaqjcm4hjqpg5fehj",
 "payment_hash": "36a46ca03b0be1294dd9c295da153e0f9cfabc98d52cd008
 367fd59f04c73cd8",
 "msatoshi": 20000,
 "amount_msat": "20000msat",
 "status": "paid",
 "pay_index": 65,
 "msatoshi_received": 20000,
 "amount_received_msat": "20000msat",
 "paid_at": 1686121129,
 "payment_preimage": "9caf59003cb98a5f74e3105618ab751c5a3a173680e
 5634d5f7475c1d446a613",
 "description": "description bread",
 "expires_at": 1686150128
},
{
 "label": "invoice17...",
 "bolt11": "lntb200n1pjgqfkqsp552unkatutgtkz76n4wzkhsjtav8qgtf2we93dn
 7m6adndjkwyyzspp5xuz44xjs68jq4usxce724gm3gar7wfl5kqdmvptzggkfl7hx
 2xusdquv3jhxcmjd9c8g6t0dcsxyun9v9jqxqrafscqp2rzjqfcxsh9gr28y6ngph
 mk90q05ejfydpq89tjjc5rl36lfmtcv424hkffwrcqqqwsqqyqqqqlgqqqqqqgq2q
 9qyysgqc7k6vkhnapcrtadcd5ta9mqctj7sg8e67t78en290setg3s3z5tpxrvuad
 qkewn8mg4rv20twjz5jfelsxa305sdqqr02pnuc8yyxgqq9gappm",
 "payment_hash": "37055a9a50d1e40af206c67caaa3714747e727f4b01
 bb60562422c9ffae651b9",
 "msatoshi": 20000,
 "amount_msat": "20000msat",
 "status": "paid",
 "pay_index": 66,
 "msatoshi_received": 20000,
 "amount_received_msat": "20000msat",
 "paid_at": 1686121133,
 "payment_preimage": "201bd915a1138a50f95f410a6ee1c664b3b1519cc574
 546782bf91698e71ec90",
 "description": "description bread",
 "expires_at": 1686150128
},
{
 "label": "invoice18...",
 "bolt11": "lntb200n1pjgqfkpsp5en3c6q27znc7xlcduzteqn5kju344vnkj45e
 00eaky7pdvuwmhpqpp5u7rkmc8qg3d2f370u5cdy4kfkwvhyyec9fqlqn5eaaahel
 k8tnaqdquv3jhxcmjd9c8g6t0dcsxyun9v9jqxqrafscqp2rzjqfcxsh9gr28y6ng
 phmk90q05ejfydpq89tjjc5rl36lfmtcv424hkffwrcqqqwsqqyqqqqlgqqqqq
 qgq2q9qyysgqq5p97fdxp2vm65jvptdjd9kndrhp7gfh6y8jmvn9uhmwt8sed0q
 zx6frczkpwk7ccx7vlzcat2ws0hr4gq3rsd6v70h6darlfa40r0sprvgph8",
 "payment_hash": "e7876de0e0445aa4c7cfe530d256c9b3997213382a4
 1f04e99ef7b7cfec75cfa",
 "msatoshi": 20000,
 "amount_msat": "20000msat",
 "status": "paid",
 "pay_index": 67,
 "msatoshi_received": 20000,
 "amount_received_msat": "20000msat",
 "paid_at": 1686121136,
 "payment_preimage": "63431b27ed5f870a2dbec4e8e52abf8cc0e30c914d68
 ba20768cc3ccdd307ba1",
 "description": "description bread",
 "expires_at": 1686150129
},
{
 "label": "invoice19...",
 "bolt11": "lntb200n1pjgqfkpsp5yknvxj76fc7adwtk6cru72eawkdf96ay6gx
 u8th70slwthus702spp5xu3r5f9u9z4rwawk2qtv2660kwzc0xzr42n4eftx060kgy9
 ves6sdquv3jhxcmjd9c8g6t0dcsxyun9v9jqxqrafscqp2rzjqfcxsh9gr28y6n
 gphmk90q05ejfydpq89tjjc5rl36lfmtcv424hkffwrcqqqwsqqyqqqqlgqqqqq
 qgq2q9qyysgqaznld7e0g2wex89s6f2l8tekss4c63hwzpa7avl6dedfy3ccw7g
 sdy46nyawj95wm8snq2clcmgvnpx9kstfrsl7uf5z5tvt72n7d4cqf67yj6",
 "payment_hash": "37223a24bc28aa3775d65016c56b4fb385879843aaa7
 5ca5667e9f6410accc35",
 "msatoshi": 20000,
 "amount_msat": "20000msat",
 "status": "paid",
 "pay_index": 68,
 "msatoshi_received": 20000,
 "amount_received_msat": "20000msat",
 "paid_at": 1686121140,
 "payment_preimage": "254a0adad61b63804a27a5dce085b5b0e9850ed85b25
 f36cd955bc52f378a6c3",
 "description": "description bread",
 "expires_at": 1686150129
},
{
 "label": "invoice20...",
 "bolt11": "lntb200n1pjgqfkpsp5upyzuaxyxlwkux8q0vc6rcgveru20sz2tfsz3w6
 qc2m5vv5tmegspp5xz09suc9yx695kulkrruwly6dqwnf9u8495vg9cc7gqnrnax
 fewqdquv3jhxcmjd9c8g6t0dcsxyun9v9jqxqrafscqp2rzjqfcxsh9gr28y6ngp
 hmk90q05ejfydpq89tjjc5rl36lfmtcv424hkffwrcqqqwsqqyqqqqlgqqqqqqgq
 2q9qyysgqj5lxwgmhrd70skwykh5mar7xr88cay0ryymermujdce8h3dqpdgqkzu
 pz729x5maxhf504yxy0hzqva6g0qyk7s7u5mvs3vs4gqkrpcqjpnpya",
 "payment_hash": "309e58730521b45a5b9fb0c7c77c9a681d349787a9
 68c41718f20131cfa64e5c",
 "msatoshi": 20000,
 "amount_msat": "20000msat",
 "status": "paid",
 "pay_index": 69,
 "msatoshi_received": 20000,
 "amount_received_msat": "20000msat",
 "paid_at": 1686121143,
 "payment_preimage": "ae36dacd7ae7cf7e5fc9259d1e450655b827da764c1
 581090a307f012493eb4d",
 "description": "description bread",
 "expires_at": 1686150129
},
{
 "label": "invoice21...",
 "bolt11": "lntb200n1pjgqfkpsp5cepcdfwgt67hf5lhxstfe7ne0pdddwun9eqezpr
 2g28geg946arspp5xar2aeqx9rtj7v3pur4tqcw3l0sx67lmqrwypg0shutpkagz8
 w2qdquv3jhxcmjd9c8g6t0dcsxyun9v9jqxqrafscqp2rzjqfcxsh9gr28y6ngphm
 k90q05ejfydpq89tjjc5rl36lfmtcv424hkffwrcqqqwsqqyqqqqlgqqqqqqgq2q9
 qyysgqptnjz8z8le0pykkjapd5y50pm7ha5rwpj5vyppq6a6m7t7zu7hwkae2umk6
 6rqcn4cl5g79wmyquxnf2h45uqnyvf46rg0m2vvnmwhsphyhup0",
 "payment_hash": "3746aee40628d72f3221e0eab061d1fbe06d7bfb00dc40a1f0bf
 161b75023b94",
 "msatoshi": 20000,
 "amount_msat": "20000msat",
 "status": "paid",
 "pay_index": 70,
 "msatoshi_received": 20000,
 "amount_received_msat": "20000msat",
 "paid_at": 1686121146,
 "payment_preimage": "c9a4ed9314fcec14d01997f9c6f9aae62a433c750
 c971a822a978ff3c65e553f",
 "description": "description bread",
 "expires_at": 1686150129
},
{
 "label": "invoice22...",
 "bolt11": "lntb200n1pjgqfkpsp5qq4qr47vd6j0nx9n2pfppp27tnad6k6jr5zcc
 jmc6g7yz2kpaqhqpp5m9cmhfzpjfg496lzuts423kt9rm6uslzpfx5ppj33k62c4x2k
 e2qdquv3jhxcmjd9c8g6t0dcsxyun9v9jqxqrafscqp2rzjqfcxsh9gr28y6ngphmk9
 0q05ejfydpq89tjjc5rl36lfmtcv424hkffwrcqqqwsqqyqqqqlgqqqqqqgq2q9qyy
 sgq9smpzq7ajw3akfw590a2aqw4wqng9hdd997d43kg4c05v7tpvzn9twk0rnutzp2
 sl5m63uk6qknrh5heyd45k6ghylrjavk7dnrvr8gqtv5vuz",
 "payment_hash": "d971bba441925152ebe2e2e15546cb28f7ae43e20a4d40865
 18db4ac54cab654",
 "msatoshi": 20000,
 "amount_msat": "20000msat",
 "status": "paid",
 "pay_index": 71,
 "msatoshi_received": 20000,
 "amount_received_msat": "20000msat",
 "paid_at": 1686121150,
 "payment_preimage": "8e3cbf15877c1134aff95e7b47190e557f2c806db843
 dd23ea65fdeec1b383b9",
 "description": "description bread",
 "expires_at": 1686150129
},
{
 "label": "invoice23...",
 "bolt11": "lntb200n1pjgqfkpsp5nfr32523jvuq2jyyd6dk8npu5600hhs9t9vzr
 94llnrsqe76fpmspp5nmzutdwryk74evhga0p9l5kegw04wn69lqv8ee0qr77pwp
 lqvemsdquv3jhxcmjd9c8g6t0dcsxyun9v9jqxqrafscqp2rzjqfcxsh9gr28y6n
 gphmk90q05ejfydpq89tjjc5rl36lfmtcv424hkffwrcqqqwsqqyqqqqlgqqqqqq
 gq2q9qyysgqzh256ztmk8uckz3ml7ru0nhtlkp3kr5gq9yk4lgy6lvzeqye20gp8k
 yc4cfh3r9ehrfl43fdwqpj97d77fxwfudky62pfjhd32h8ruspsejrzy",
 "payment_hash": "9ec5c5b5c325bd5cb2e8ebc25fd2d9439f574f45f
 8187ce5e01fbc1707e06677",
 "msatoshi": 20000,
 "amount_msat": "20000msat",
 "status": "paid",
 "pay_index": 72,
 "msatoshi_received": 20000,
 "amount_received_msat": "20000msat",
 "paid_at": 1686121153,
 "payment_preimage": "8822160edf8f1816c23505b9d28a9b5da8c3291d04
 f94217d3e75d26693d579d",
 "description": "description bread",
 "expires_at": 1686150129
},
{
 "label": "invoice24...",
 "bolt11": "lntb200n1pjgqfkpsp539xernshmaus477x0yfya5eh5uyw7tsdmxr82jtde
 2ezmac58lxqpp52wmc8pql0z0ux3nj7e0eu27z5uufgqsqnavf43vfgs6fnprfwum
 qdquv3jhxcmjd9c8g6t0dcsxyun9v9jqxqrafscqp2rzjqfcxsh9gr28y6ngphmk9
 0q05ejfydpq89tjjc5rl36lfmtcv424hkffwrcqqqwsqqyqqqqlgqqqqqqgq2q9qy
 ysgqgjy2jtagfq9wlkwfahy4dssukajalj0m05fnk60xfvq8h2myqtqrya6xmtaex
 f28zngtwgf3n9k382k0ctxealutu6a55kzdtq8qf0qqkals8c",
 "payment_hash": "53b783841f789fc34672f65f9e2bc2a7389402009f589ac5894434
 9984697736",
 "msatoshi": 20000,
 "amount_msat": "20000msat",
 "status": "paid",
 "pay_index": 73,
 "msatoshi_received": 20000,
 "amount_received_msat": "20000msat",
 "paid_at": 1686121157,
 "payment_preimage": "59f596f7d8e9a547f1797b2f23c732b2fd436f59c86
 1e1cc2dde0eeb6e6befeb",
 "description": "description bread",
 "expires_at": 1686150129
},
{
 "label": "invoice25...",
 "bolt11": "lntb200n1pjgqfkzsp5gwms62zpchhnj720a93c40ktjjt2lg9fgyeahfg5
 z0kxl00fymkspp5hmus2zm4krk8gxyjh5snk9fkttrs0tg5lm565mtw897mj0p6ed
 2qdquv3jhxcmjd9c8g6t0dcsxyun9v9jqxqrafscqp2rzjqfcxsh9gr28y6ngphmk
 90q05ejfydpq89tjjc5rl36lfmtcv424hkffwrcqqqwsqqyqqqqlgqqqqqqgq2q9q
 yysgqezsnf79yvd6r8qfxfja6geg736pff8xctnq7yldjn399f9x4yu25fmjjky4d
 uxcmmjvehlcurygz3pm54sucsefahz9g0kr9l8q4nxspswn78g",
 "payment_hash": "bef9050b75b0ec741892bd213b15365ac707ad14fee9aa6d6e397
 db93c3acb54",
 "msatoshi": 20000,
 "amount_msat": "20000msat",
 "status": "paid",
 "pay_index": 74,
 "msatoshi_received": 20000,
 "amount_received_msat": "20000msat",
 "paid_at": 1686121160,
 "payment_preimage": "59da17d9ad54617b37dcda6e6a55a06668dcfa401a416
 b45e6c7ea6c85a288b7",
 "description": "description bread",
 "expires_at": 1686150130
},
{
 "label": "invoice26...",
 "bolt11": "lntb200n1pjgqfkzsp5ratkjucyx5en0egg2zdcgl03rys3zeaxgac
 5kcmrur7qf7yqepvqpp53zg8tk2cnja90rwgahkxzn8p2tcewnxf7r4lxnlake75p0
 wftvdqdquv3jhxcmjd9c8g6t0dcsxyun9v9jqxqrafscqp2rzjqfcxsh9gr28y6ng
 phmk90q05ejfydpq89tjjc5rl36lfmtcv424hkffwrcqqqwsqqyqqqqlgqqqqqqg
 q2q9qyysgq3g83jzulrfjtj30wn6tdr4vd6p5t3hjdvq03tpffc75wna9xh5wjf0
 jvvtkgermz6ur4x54vzqxcqaumw5ftqnjjzy3e47zgky3hpvgpyf6jr9",
 "payment_hash": "889075d9589cba578dc8edec614ce152f1974cc9f0eb
 f34ffdb67d40bdc95b1a",
 "msatoshi": 20000,
 "amount_msat": "20000msat",
 "status": "paid",
 "pay_index": 75,
 "msatoshi_received": 20000,
 "amount_received_msat": "20000msat",
 "paid_at": 1686121164,
 "payment_preimage": "63eeb3b30e6d6ca6500ffffbdce5530f028181bde489
 a4479df039ff8d952fc3",
 "description": "description bread",
 "expires_at": 1686150130
},
{
 "label": "invoice27...",
 "bolt11": "lntb200n1pjgqfkzsp5znae597euwek6jmrx9huhj8hk4uvaeltr9tu
 9yj850du499u3cfqpp5q4x5kxxzdta3pze6ph84kpwszdxdhhmqe5n3hsa4plmxwkl
 jc0ssdquv3jhxcmjd9c8g6t0dcsxyun9v9jqxqrafscqp2rzjqfcxsh9gr28y6ngphm
 k90q05ejfydpq89tjjc5rl36lfmtcv424hkffwrcqqqwsqqyqqqqlgqqqqqqgq2q9
 qyysgqx3l8g268umgu4cf4qy7d0a4yt6ly3p5vd89cdywcxhwx7vwjmfrk224sfrh
 a644kuwc26drg6lwtzxkp9c2fjawxfuujrlk8r8xfw4cqxtfhj4",
 "payment_hash": "054d4b18c26afb108b3a0dcf5b05d0134cdbdf60cd2
 71bc3b50ff6675bf2c3e1",
 "msatoshi": 20000,
 "amount_msat": "20000msat",
 "status": "paid",
 "pay_index": 76,
 "msatoshi_received": 20000,
 "amount_received_msat": "20000msat",
 "paid_at": 1686121167,
 "payment_preimage": "a90a521edc6db361c2db556de7e95d75a08cacb385fe
 cf6cfa530e67ca83eb2e",
 "description": "description bread",
 "expires_at": 1686150130
},
{
 "label": "invoice28...",
 "bolt11": "lntb200n1pjgqfkzsp5z335fgl5s87nxjhlj0y3rqtshrn2wcml7jnlatr
 dj40e468qjwlqpp5cjgvf4dz6yz5ypev9va2ztcqrgwzzzttlxfhlpeawf48vh3w
 lhusdquv3jhxcmjd9c8g6t0dcsxyun9v9jqxqrafscqp2rzjqfcxsh9gr28y6ngp
 hmk90q05ejfydpq89tjjc5rl36lfmtcv424hkffwrcqqqwsqqyqqqqlgqqqqqqgq
 2q9qyysgqhufutsuhn9aggekla5paymmrj4z6hht26l59ft3agw58yy5t34dhwfs
 su2unw2545vzzc4re8j32p9pxjmgev9tmcq8hurdalpus5jgpmpgmdd",
 "payment_hash": "c490c4d5a2d10542072c2b3aa12f001a1c21096bf99
 37f873d726a765e2efdf9",
 "msatoshi": 20000,
 "amount_msat": "20000msat",
 "status": "paid",
 "pay_index": 77,
 "msatoshi_received": 20000,
 "amount_received_msat": "20000msat",
 "paid_at": 1686121170,
 "payment_preimage": "aa4153493800dcd4b7ad31f77e09a6bf4b86754269ab
 d4b4a0bb162754bb5072",
 "description": "description bread",
 "expires_at": 1686150130
},
{
 "label": "invoice29...",
 "bolt11": "lntb200n1pjgqfkzsp5msgzhchhk208cc89742vtvuwr0d34wxtazas9x0
 s60dha0uenw2qpp5hg7ccmfc38dx2xedvv50nw7pl7jh7dpnznjvqpd9cfxu5qhp
 hqmqdquv3jhxcmjd9c8g6t0dcsxyun9v9jqxqrafscqp2rzjqfcxsh9gr28y6ngp
 hmk90q05ejfydpq89tjjc5rl36lfmtcv424hkffwrcqqqwsqqyqqqqlgqqqqqqgq
 2q9qyysgqmpcht5h3m6vgu54h4esrc28ve9elf2yutf3dyhqh5e2mj597zdjsca3
 akvuyvdv992rqekvd8kg6wu200wetlw98ca6drzwghp927sqqeresmz",
 "payment_hash": "ba3d8c6d3889da651b2d6328f9bbc1ffa57f343314e4c
 005a5c24dca02e1b836",
 "msatoshi": 20000,
 "amount_msat": "20000msat",
 "status": "paid",
 "pay_index": 78,
 "msatoshi_received": 20000,
 "amount_received_msat": "20000msat",
 "paid_at": 1686121174,
 "payment_preimage": "a27c33ec47708ba91a4a56cc48d576682d7bdddd7c239
 17e370ee6bef9e8888c",
 "description": "description bread",
 "expires_at": 1686150130
},
{
 "label": "invoice30...",
 "bolt11": "lntb200n1pjgqfkzsp528le5vyvm6qhvzrfy7slg0whkgkejutdcr5t
 ull5wwtdrklv8cvqpp5aggzryhcnz0vcasx5smpxfzwkgnptfr44muq876m7x5c7aa
 a8fssdquv3jhxcmjd9c8g6t0dcsxyun9v9jqxqrafscqp2rzjqfcxsh9gr28y6ngphm
 k90q05ejfydpq89tjjc5rl36lfmtcv424hkffwrcqqqwsqqyqqqlgqqqqqqgq2q9
 qyysgqcss604uv2wu0jxrw7gecxah9lpw7tfyhrf4rnqlmcdsggf5h5derz7gahyh
 9keqg9h0eqy2f5l28p9jn6dl7jkhrkdzj63h8tyzy6ccqmz6ph9",
 "payment_hash": "ea102192f8989ecc7606a43613244eb22615a475aef803
 fb5bf1a98f77bd3a61",
 "msatoshi": 20000,
 "amount_msat": "20000msat",
 "status": "paid",
 "pay_index": 79,
 "msatoshi_received": 20000,
 "amount_received_msat": "20000msat",
 "paid_at": 1686121178,
 "payment_preimage": "827e65b605db2301069263deaace127d4ad9e1dcb4f087
 5a110fd300f51121c1",
 "description": "description bread",
 "expires_at": 1686150130
},
{
 "label": "invoice31...",
 "bolt11": "lntb200n1pjgqfkzsp5v72299xag0a485x2gg2w96ercm972yags26
 v75x5wu3ww262vzxqpp588rj6n8xjle5yxd2n2wvcucefr7hgyze3u7yf4s65h67ql
 q2qngsdquv3jhxcmjd9c8g6t0dcsxyun9v9jqxqrafscqp2rzjqfcxsh9gr28y6
 ngphmk90q05ejfydpq89tjjc5rl36lfmtcv424hkffwrcqqqwsqqyqqqlgqqqq
 qqgq2q9qyysgqrsrsl0ssr6ca42p777ufkkmwyw2t04u6ze5r9d6efjnf4l7y49
 8pdtcqma57aq6frlsa66apj42zp2qupea8v9sua04vd6zgs38g4dcqrw4r7j",
 "payment_hash": "39c72d4ce697f34219aa9a9ccc731948fd7410598f3c
 44d61aa5f5e07c0a04d1",
 "msatoshi": 20000,
 "amount_msat": "20000msat",
 "status": "paid",
 "pay_index": 80,
 "msatoshi_received": 20000,
 "amount_received_msat": "20000msat",
 "paid_at": 1686121181,
 "payment_preimage": "c4215e142444cf5ccabdf944044dc8f07908a03f8e7e
 5c4eb6568763cad98f92",
 "description": "description bread",
 "expires_at": 1686150130
},
{
 "label": "invoice32...",
 "bolt11": "lntb200n1pjgqfkzsp5nervrpugwqe568v2qzmznz3a42skaw6geqtgezdl
 dxramnt9kevqpp58y4t7uegxp4kwtevdql84fe6cajsde7kxm0fwp5hwx0d79qmz0m
 qdquv3jhxcmjd9c8g6t0dcsxyun9v9jqxqrafscqp2rzjqfcxsh9gr28y6ngphmk90
 q05ejfydpq89tjjc5rl36lfmtcv424hkffwrcqqqwsqqyqqqqlgqqqqqqgq2q9qyys
 gq9mapf53w0d22hr4v9kca8f4spx0gy7mg3284q0ffz82wlkfmh9l5hyg3t3nj2lnl
 r7fyw75wnzt5jz38qx6dl6ud90k637juhjj3cnqpemxt2m",
 "payment_hash": "392abf7328306b672f2c683e7aa73ac76506e7d636de
 970697719edf141b13f6",
 "msatoshi": 20000,
 "amount_msat": "20000msat",
 "status": "paid",
 "pay_index": 81,
 "msatoshi_received": 20000,
 "amount_received_msat": "20000msat",
 "paid_at": 1686121184,
 "payment_preimage": "9f6b8ee2487daf9ad33a007036cb10d1e7d33e24f3f1e
 f37f6a5648dfb6111e0",
 "description": "description bread",
 "expires_at": 1686150130
},
{
 "label": "invoice33...",
 "bolt11": "lntb200n1pjgqfkrsp52ljuw5ts0hmz25alzykrgxsgd3pp0n8ngtsh9fxn
 cxhfur5amskqpp5xkq9jjjxkud22mxgmxlnhp4944v0tx9ms8a076xpnggrss9f5n
 esdquv3jhxcmjd9c8g6t0dcsxyun9v9jqxqrafscqp2rzjqfcxsh9gr28y6ngphmk
 90q05ejfydpq89tjjc5rl36lfmtcv424hkffwrcqqqwsqqyqqqqlgqqqqqqgq2q9q
 yysgq6l294qwq9z9samqw2kd8x8vs3gl5787fv78kk2qs4r8t9t0smlmsanlfhq2a
 y6dgt4c8hve3rwt8249er6zmmmq8h7y08pnarft95msqu9lw3r",
 "payment_hash": "3580594a46b71aa56cc8d9bf3b86a5ad58f598bb81faf
 f68c19a103840a9a4f3",
 "msatoshi": 20000,
 "amount_msat": "20000msat",
 "status": "paid",
 "pay_index": 82,
 "msatoshi_received": 20000,
 "amount_received_msat": "20000msat",
 "paid_at": 1686121188,
 "payment_preimage": "4d428a424a5e33e046d73df36ca2c9c06e1969c89065c
 3b668fadf9bb59dd4ac",
 "description": "description bread",
 "expires_at": 1686150131
},
{
 "label": "invoice34...",
 "bolt11": "lntb200n1pjgqfkrsp5q2v4upk0sh97apu26gh3pnnwhcmsn0xaqzyd88
 qr55yyus4k5xcspp5and36g7efwehy7980j7g3d744fr3vn4mrjnuzzpcyht3t
 qgesgesdquv3jhxcmjd9c8g6t0dcsxyun9v9jqxqrafscqp2rzjqfcxsh9gr28
 y6ngphmk90q05ejfydpq89tjjc5rl36lfmtcv424hkffwrcqqqwsqqyqqqqlgq
 qqqqqgq2q9qyysgqyv35nuqnz9dc64zzk0neygunp466gr3c0hntwffyfdp8pz5
 kew09nv52ll7dzk957nfgwg5s5nj3heewr5qnjpmkzmpv5uvpe780kmsq8er237",
 "payment_hash": "ecdb1d23d94bb37278a77cbc88b7d5aa47164ebb1ca
 7c1083825d71581198233",
 "msatoshi": 20000,
 "amount_msat": "20000msat",
 "status": "paid",
 "pay_index": 83,
 "msatoshi_received": 20000,
 "amount_received_msat": "20000msat",
 "paid_at": 1686121191,
 "payment_preimage": "2b4f435c65d796ce17da2a52cefaddd95a605a8e90f5
 e61d5d5c52b9b1f46b2c",
 "description": "description bread",
 "expires_at": 1686150131
},
{
 "label": "invoice35...",
 "bolt11": "lntb200n1pjgqfkrsp52ltuxm3ne3dfkukkz8av6h9xux3nutdpjwc
 llp5uxkc8uymxfnuspp5k5zsazewery68y5scj6dzsv74j020v57genja8n4e9styng
 tdvkqdquv3jhxcmjd9c8g6t0dcsxyun9v9jqxqrafscqp2rzjqfcxsh9gr28y6ngphmk
 90q05ejfydpq89tjjc5rl36lfmtcv424hkffwrcqqqwsqqyqqqqlgqqqqqqgq2q9q
 yysgq4wnlwln8u35xqj4n2z52urv4xcswk0zex2sdp3sa56pdyn00wm3qjntrf64c
 7mu74es9zr97vfya4mhha6a3xhsmwqwyml5r7q526hsqe9jnpn",
 "payment_hash": "b5050e8b2ec8c9a39290c4b4d1419eac9ea7b29e466
 72e9e75c960b24d0b6b2c",
 "msatoshi": 20000,
 "amount_msat": "20000msat",
 "status": "paid",
 "pay_index": 84,
 "msatoshi_received": 20000,
 "amount_received_msat": "20000msat",
 "paid_at": 1686121195,
 "payment_preimage": "a1cb313335c46249baddf01e0474b4afd0ab0af96633
 ed2ec09da51c56ed1c04",
 "description": "description bread",
 "expires_at": 1686150131
},
{
 "label": "invoice36...",
 "bolt11": "lntb200n1pjgqfkrsp5g5vmvg7argvzdey4y5djv5d6a2dknqzk8yhe0rpf
 8lptsnav3lqspp5tdmscncrvv8d0v4pfdrexndhlv7eqz8lqhlp820ldvqwd7cwvsr
 sdquv3jhxcmjd9c8g6t0dcsxyun9v9jqxqrafscqp2rzjqfcxsh9gr28y6ngphmk9
 0q05ejfydpq89tjjc5rl36lfmtcv424hkffwrcqqqwsqqyqqqqlgqqqqqqgq2q9qy
 ysgqt7vh3das3vylhxepcnqjw5rswsjydgk6aguvpyf2npwvkktcqq89exvz0mrph
 xpg7y0pm9espr38yqz6kuxe4rsdx6mda063q3ss3tqqtqplkg",
 "payment_hash": "5b770c4f03630ed7b2a14b47934db7fb3d9008ff05fe13
 a9ff6b00e6fb0e6407",
 "msatoshi": 20000,
 "amount_msat": "20000msat",
 "status": "paid",
 "pay_index": 85,
 "msatoshi_received": 20000,
 "amount_received_msat": "20000msat",
 "paid_at": 1686121198,
 "payment_preimage": "6751e20787c2d3993f220119c040fcc1e6f6ff06c336c7e
 dd28fd7bb47ff761b",
 "description": "description bread",
 "expires_at": 1686150131
},
{
 "label": "invoice37...",
 "bolt11": "lntb200n1pjgqfkrsp5l78zqkvmvcve5w8c4akyrsmvzs9ff7k5cg0akd
 j8hzekgdssjzjqpp5jtg37kc86aj6thpxs2rpss0w855d6m4q5trlxadx8p3v69v
 8fmesdquv3jhxcmjd9c8g6t0dcsxyun9v9jqxqrafscqp2rzjqfcxsh9gr28y6ng
 phmk90q05ejfydpq89tjjc5rl36lfmtcv424hkffwrcqqqwsqqyqqqqlgqqqqqqg
 q2q9qyysgqy7mrwwsdllczzlukh4vs85l0900eajrpfvzq0vkz0c30ey62se25k4
 6pd9704sv9d6cul9p35l72n8wv4n5qjvgjy68a9wxy2dpmqqcqn3hzev",
 "payment_hash": "92d11f5b07d765a5dc2682861841ee3d28dd6ea0a2c7f3
 75a63862cd15874ef3",
 "msatoshi": 20000,
 "amount_msat": "20000msat",
 "status": "paid",
 "pay_index": 86,
 "msatoshi_received": 20000,
 "amount_received_msat": "20000msat",
 "paid_at": 1686121201,
 "payment_preimage": "675f2ca7292ea32cb826990b7ce1faaed5395c23386592
 f3cd444bc3804854c5",
 "description": "description bread",
 "expires_at": 1686150131
},
{
 "label": "invoice38...",
 "bolt11": "lntb200n1pjgqfkrsp5mwvnl389dv4mfaast9jnmmgwvs4f83ps8pxaj
 ga7sz4kz5g9exnqpp5f88vndh4jkh868hlxkd98rsue6w8axta3v0ph55v8za3us3x
 dw8qdquv3jhxcmjd9c8g6t0dcsxyun9v9jqxqrafscqp2rzjqfcxsh9gr28y6ng
 phmk90q05ejfydpq89tjjc5rl36lfmtcv424hkffwrcqqqwsqqyqqqqlgqqqqqqg
 q2q9qyysgq8lfy2a0450c3fmdrzt5nefgvwh9mqjfu3zf0sxarjykh09eulgjnrg
 mlpxu9kdfvv9sqf4lhjcn80zkwhkc542rc5407vr65yp6352gpapf7nw",
 "payment_hash": "49cec9b6f595ae7d1eff359a538e1cce9c7e997d8b1
 e1bd28c38bb1e42266b8e",
 "msatoshi": 20000,
 "amount_msat": "20000msat",
 "status": "paid",
 "pay_index": 87,
 "msatoshi_received": 20000,
 "amount_received_msat": "20000msat",
 "paid_at": 1686121206,
 "payment_preimage": "1575e47aebbf7f46ce1033383a31fdbac3571af5b2d
 a4e57062067156fefc78e",
 "description": "description bread",
 "expires_at": 1686150131
},
{
 "label": "invoice39...",
 "bolt11": "lntb200n1pjgqfkrsp5pmwkrfqz3jk5h0dy4w6xljjk5zfeaj5sy3076a
 txaqmdjkl38kkqpp5x75yxu5kcxe09zkhs8xxxaunjlur6rjdsead4vujrt7y560k5cw
 qdquv3jhxcmjd9c8g6t0dcsxyun9v9jqxqrafscqp2rzjqfcxsh9gr28y6ngphmk9
 0q05ejfydpq89tjjc5rl36lfmtcv424hkffwrcqqqwsqqyqqqqlgqqqqqqgq2q9qy
 ysgqhdpm6djas3eenh2nv3zn2mcdzg3wsmdcrk430u703p6e490wmvuk8hfgn3ep5
 wxg4t6vg9jxap7klh355wasth8prum5ul5jmk88w6cpamv0hn",
 "payment_hash": "37a8437296c1b2f28ad781cc63779397f83d0e4d867
 adab3921afc4a69f6a61c",
 "msatoshi": 20000,
 "amount_msat": "20000msat",
 "status": "paid",
 "pay_index": 88,
 "msatoshi_received": 20000,
 "amount_received_msat": "20000msat",
 "paid_at": 1686121209,
 "payment_preimage": "1c4a17b8134f65ef24ebde970683dc85238c03cac46a
 7499ea010ee62ed68df7",
 "description": "description bread",
 "expires_at": 1686150131
},
{
 "label": "invoice40...",
 "bolt11": "lntb200n1pjgqfkrsp5phj85r2pl6dstpdj87krrmyd50wae0src3dj0aq6
 9zsux66df7qspp5x0rvrxem89wjtsxmryu6nq5r2dpj0r090eqkjvm2v99sn78s220
 sdquv3jhxcmjd9c8g6t0dcsxyun9v9jqxqrafscqp2rzjqfcxsh9gr28y6ngphmk90
 q05ejfydpq89tjjc5rl36lfmtcv424hkffwrcqqqwsqqyqqqqlgqqqqqqgq2q9qyys
 gq4q6tdpjlyzlvdwv6mdppzk3mm220mrmcjlzjr5k2n9yk4x7pjz8sf8pn7qpm49nx
 6fvfmt8wrfgnztxvp2fyez60egwhtjggd4xw3dgqg8t4jx",
 "payment_hash": "33c6c19b3b395d25c0db1939a982835343278de57e4169
 336a614b09f8f0529f",
 "msatoshi": 20000,
 "amount_msat": "20000msat",
 "status": "paid",
 "pay_index": 89,
 "msatoshi_received": 20000,
 "amount_received_msat": "20000msat",
 "paid_at": 1686121213,
 "payment_preimage": "1b4b40567484c404c4cd1392f8a5ab4ee767ba61b7e6e6
 ce52b41beef6756f96",
 "description": "description bread",
 "expires_at": 1686150131
}
```

Listing A.12 Node 2 payments of Node 1 invoices

```
{
   "destination": "032632259765dd04833258446729d6ef7835c0a7b2d
   cdf3dcc407d09a477a05ab1",
   "payment_hash": "127d97d9c2800cc29c50c52189015392db5f31a8e17
   d689762cbe46e489ab5fd",
   "created_at": 1686121083.181,
    "parts": 1,
    "msatoshi": 20000,
    "amount_msat": "20000msat",
    "msatoshi_sent": 21000,
    "amount_sent_msat": "21000msat",
    "payment_preimage": "16ea1c70df51426ab0f6dc0c2b8047758824d9498b7
    f6eaffef7cabf6a362c02",
    "status": "complete"
}
{
    "destination": "032632259765dd04833258446729d6ef7835c0a7b2d
    cdf3dcc407d09a477a05ab1",
    "payment_hash": "08b7fecd340930c411517a2f76094ff2a3eb547f547
    c7e70d2eeb912c517eadf",
    "created_at": 1686121085.975,
    "parts": 1,
    "msatoshi": 20000,
    "amount_msat": "20000msat",
    "msatoshi_sent": 21000,
    "amount_sent_msat": "21000msat",
    "payment_preimage": "53dff9e1a6708356aeca5e80c79864ed248f9e7b42c
    543f7b95555062a2adaa2",
    "status": "complete"
}
{
    "destination": "032632259765dd04833258446729d6ef7835c0a7b2d
    cdf3dcc407d09a477a05ab1",
    "payment_hash": "0874dbb059b0a2a7614cea6a7f3a70023803a614196
    787ed0a32791564bf762e",
    "created_at": 1686121089.405,
    "parts": 1,
    "msatoshi": 20000,
    "amount_msat": "20000msat",
    "msatoshi_sent": 21000,
    "amount_sent_msat": "21000msat",
    "payment_preimage": "812ea012310475c005991ef1b272bb64efbecfd469f
    a169d0d577bced0e3eb72",
    "status": "complete"
}
{
    "destination": "032632259765dd04833258446729d6ef7835c0a7b2d
    cdf3dcc407d09a477a05ab1",
    "payment_hash": "6e4966a2ee9193d4c48f943170b24be017e2a5738c1
    0646765738fb93189a7d3",
    "created_at": 1686121092.858,
    "parts": 1,
    "msatoshi": 20000,
    "amount_msat": "20000msat",
    "msatoshi_sent": 21000,
    "amount_sent_msat": "21000msat",
    "payment_preimage": "813ac2c9c8d7e2f0ed908cd3c9faeaf217f80deed68
    b5e20b2ec2bb31e3328f6",
    "status": "complete"
}
{
    "destination": "032632259765dd04833258446729d6ef7835c0a7b2d
    cdf3dcc407d09a477a05ab1",
    "payment_hash": "ecd7e73456014d21213b5548a4a62e17f8a9579ad82f
    71492ae66f007d884c70",
    "created_at": 1686121096.398,
    "parts": 1,
    "msatoshi": 20000,
    "amount_msat": "20000msat",
    "msatoshi_sent": 21000,
    "amount_sent_msat": "21000msat",
    "payment_preimage": "40d5b91bcd60886580baa83ca83f779c39975429686c
    8fcc79799c0ac9674d57",
    "status": "complete"
}
{
    "destination": "032632259765dd04833258446729d6ef7835c0a7b2dc
    df3dcc407d09a477a05ab1",
    "payment_hash": "6b411bc5c7fb46d49fda2754a2c9007d7d0ae020feee
    c166a28d2d7f921f5ebc",
    "created_at": 1686121099.544,
    "parts": 1,
    "msatoshi": 20000,
    "amount_msat": "20000msat",
    "msatoshi_sent": 21000,
    "amount_sent_msat": "21000msat",
    "payment_preimage": "11bf5cbf969a5f9dec4e59410c7896ba2c2e809a763c
    05b34310f1f076f1ca0f",
    "status": "complete"
}
{
    "destination": "032632259765dd04833258446729d6ef7835c0a7b2dc
    df3dcc407d09a477a05ab1",
    "payment_hash": "d1fb471494ae03d74c81c58ed1dfc6b4eef1cd3ec65f
    f1bbcc09f81b2bc0c8a4",
    "created_at": 1686121103.110,
    "parts": 1,
    "msatoshi": 20000,
    "amount_msat": "20000msat",
    "msatoshi_sent": 21000,
    "amount_sent_msat": "21000msat",
    "payment_preimage": "f831749594a3ec55c45921e9cd9cd962442b7f5815f5
    a183a58f9767952c59a5",
    "status": "complete"
}
{
    "destination": "032632259765dd04833258446729d6ef7835c0a7b2dc
    df3dcc407d09a477a05ab1",
    "payment_hash": "86259fae2cf6ccc14f223db8b12655aad9fcaed64ed5
    4c0b1f79b1ab6bb02c19",
    "created_at": 1686121106.389,
    "parts": 1,
    "msatoshi": 20000,
    "amount_msat": "20000msat",
    "msatoshi_sent": 21000,
    "amount_sent_msat": "21000msat",
    "payment_preimage": "d3b25afd61350da805b5ad025170c4c5d6a41afa13e
    83c66da796dc223327d67",
    "status": "complete"
}
{
    "destination": "032632259765dd04833258446729d6ef7835c0a7b2dc
    df3dcc407d09a477a05ab1",
    "payment_hash": "0b4b1e57e0128bd33e6e81f1b5beb0c603b81046e23
    d346ed4c3e0391f1ab980",
    "created_at": 1686121109.802,
    "parts": 1,
    "msatoshi": 20000,
    "amount_msat": "20000msat",
    "msatoshi_sent": 21000,
    "amount_sent_msat": "21000msat",
    "payment_preimage": "d806e7caa499a76fdae553e8390866f15281d600e58
    25ccfde9ee6d93da0489d",
    "status": "complete"
}
{
    "destination": "032632259765dd04833258446729d6ef7835c0a7b2d
    cdf3dcc407d09a477a05ab1",
    "payment_hash": "89e9d1bfacf83e43d66d87adee1cfb194cb0708edc6a
    477a1dbefcf00bde70df",
    "created_at": 1686121113.250,
    "parts": 1,
    "msatoshi": 20000,
    "amount_msat": "20000msat",
    "msatoshi_sent": 21000,
    "amount_sent_msat": "21000msat",
    "payment_preimage": "dfb582b416fce7fb9f16da8e6b4bd6c6d1b16c079c5b
    83abe8b2d9c416b6cb6b",
    "status": "complete"
}
{
    "destination": "032632259765dd04833258446729d6ef7835c0a7b2dcd
    f3dcc407d09a477a05ab1",
    "payment_hash": "323e3bb6fcdb456b48bb01c01a0a106e5fdf8e7eff28
    081dfe8f3303f2b3d396",
    "created_at": 1686121116.579,
    "parts": 1,
    "msatoshi": 20000,
    "amount_msat": "20000msat",
    "msatoshi_sent": 21000,
    "amount_sent_msat": "21000msat",
    "payment_preimage": "79a2ab0b3cd10c88920b68de974bdd24b9ddfe900ba5e
    41c7afb6074ff8cf4c3",
    "status": "complete"
}
{
    "destination": "032632259765dd04833258446729d6ef7835c0a7b2dcd
    f3dcc407d09a477a05ab1",
    "payment_hash": "f05c783824515c3aa8b1fbf1b7a0a19391fb072c319f7
    57a63c26f45c8298e3c",
    "created_at": 1686121119.880,
    "parts": 1,
    "msatoshi": 20000,
    "amount_msat": "20000msat",
    "msatoshi_sent": 21000,
    "amount_sent_msat": "21000msat",
    "payment_preimage": "611e089ca2ecdbfd2762bcc76087d7dfdf927d092839
    09ade6b088fae01ac693",
    "status": "complete"
}
{
    "destination": "032632259765dd04833258446729d6ef7835c0a7b2dc
    df3dcc407d09a477a05ab1",
    "payment_hash": "c7dda2045ecaddaf3105b93d7b231bf86e82f6389eab
    5c9cea9f224773f42cd1",
    "created_at": 1686121123.323,
    "parts": 1,
    "msatoshi": 20000,
    "amount_msat": "20000msat",
    "msatoshi_sent": 21000,
    "amount_sent_msat": "21000msat",
    "payment_preimage": "e6e7008083e0b53b527573b268ad7396f62d3c1941960
    454b3613ed61a75fb40",
    "status": "complete"
}
{
    "destination": "032632259765dd04833258446729d6ef7835c0a7b2
    dcdf3dcc407d09a477a05ab1",
    "payment_hash": "36a46ca03b0be1294dd9c295da153e0f9cfabc98d52
    cd008367fd59f04c73cd8",
    "created_at": 1686121126.645,
    "parts": 1,
    "msatoshi": 20000,
    "amount_msat": "20000msat",
    "msatoshi_sent": 21000,
    "amount_sent_msat": "21000msat",
    "payment_preimage": "9caf59003cb98a5f74e3105618ab751c5a3a173680e
    5634d5f7475c1d446a613",
    "status": "complete"
}
{
    "destination": "032632259765dd04833258446729d6ef7835c0a7b2
    dcdf3dcc407d09a477a05ab1",
    "payment_hash": "37055a9a50d1e40af206c67caaa3714747e727f4b0
    1bb60562422c9ffae651b9",
    "created_at": 1686121130.234,
    "parts": 1,
    "msatoshi": 20000,
    "amount_msat": "20000msat",
    "msatoshi_sent": 21000,
    "amount_sent_msat": "21000msat",
    "payment_preimage": "201bd915a1138a50f95f410a6ee1c664b3b1519cc5
    74546782bf91698e71ec90",
    "status": "complete"
}
{
    "destination": "032632259765dd04833258446729d6ef7835c0a7b2d
    cdf3dcc407d09a477a05ab1",
    "payment_hash": "e7876de0e0445aa4c7cfe530d256c9b3997213382a4
    1f04e99ef7b7cfec75cfa",
    "created_at": 1686121133.639,
    "parts": 1,
    "msatoshi": 20000,
    "amount_msat": "20000msat",
    "msatoshi_sent": 21000,
    "amount_sent_msat": "21000msat",
    "payment_preimage": "63431b27ed5f870a2dbec4e8e52abf8cc0e30c914d
    68ba20768cc3ccdd307ba1",
    "status": "complete"
}
{
    "destination": "032632259765dd04833258446729d6ef7835c0a7b
    2dcdf3dcc407d09a477a05ab1",
    "payment_hash": "37223a24bc28aa3775d65016c56b4fb385879843aa
    a75ca5667e9f6410accc35",
    "created_at": 1686121137.047,
    "parts": 1,
    "msatoshi": 20000,
    "amount_msat": "20000msat",
    "msatoshi_sent": 21000,
    "amount_sent_msat": "21000msat",
    "payment_preimage": "254a0adad61b63804a27a5dce085b5b0e9850ed85b
    25f36cd955bc52f378a6c3",
    "status": "complete"
}
{
    "destination": "032632259765dd04833258446729d6ef7835c0a7b2
    dcdf3dcc407d09a477a05ab1",
    "payment_hash": "309e58730521b45a5b9fb0c7c77c9a681d349787a96
    8c41718f20131cfa64e5c",
    "created_at": 1686121140.613,
    "parts": 1,
    "msatoshi": 20000,
    "amount_msat": "20000msat",
    "msatoshi_sent": 21000,
    "amount_sent_msat": "21000msat",
    "payment_preimage": "ae36dacd7ae7cf7e5fc9259d1e450655b827da764c15
    81090a307f012493eb4d",
    "status": "complete"
}
```

```
{
    "destination": "032632259765dd04833258446729d6ef7835c0a7b2
    dcdf3dcc407d09a477a05ab1",
    "payment_hash": "3746aee40628d72f3221e0eab061d1fbe06d7bfb00
    dc40a1f0bf161b75023b94",
    "created_at": 1686121143.967,
    "parts": 1,
    "msatoshi": 20000,
    "amount_msat": "20000msat",
    "msatoshi_sent": 21000,
    "amount_sent_msat": "21000msat",
    "payment_preimage": "c9a4ed9314fcec14d01997f9c6f9aae62a433c750c
    971a822a978ff3c65e553f",
    "status": "complete"
}
{
    "destination": "032632259765dd04833258446729d6ef7835c0a7b
    2dcdf3dcc407d09a477a05ab1",
    "payment_hash": "d971bba441925152ebe2e2e15546cb28f7ae43e20
    a4d4086518db4ac54cab654",
    "created_at": 1686121147.400,
    "parts": 1,
    "msatoshi": 20000,
    "amount_msat": "20000msat",
    "msatoshi_sent": 21000,
    "amount_sent_msat": "21000msat",
    "payment_preimage": "8e3cbf15877c1134aff95e7b47190e557f2c806db8
    43dd23ea65fdeec1b383b9",
    "status": "complete"
}
{
    "destination": "032632259765dd04833258446729d6ef7835c0a7b
    2dcdf3dcc407d09a477a05ab1",
    "payment_hash": "9ec5c5b5c325bd5cb2e8ebc25fd2d9439f574f45f
    8187ce5e01fbc1707e06677",
    "created_at": 1686121150.506,
    "parts": 1,
    "msatoshi": 20000,
    "amount_msat": "20000msat",
    "msatoshi_sent": 21000,
    "amount_sent_msat": "21000msat",
    "payment_preimage": "8822160edf8f1816c23505b9d28a9b5da8c3291d
    04f94217d3e75d26693d579d",
    "status": "complete"
```

```
}
{
    "destination": "032632259765dd04833258446729d6ef7835c0a7
    b2dcdf3dcc407d09a477a05ab1",
    "payment_hash": "53b783841f789fc34672f65f9e2bc2a7389402
    009f589ac58944349984697736",
    "created_at": 1686121154.240,
    "parts": 1,
    "msatoshi": 20000,
    "amount_msat": "20000msat",
    "msatoshi_sent": 21000,
    "amount_sent_msat": "21000msat",
    "payment_preimage": "59f596f7d8e9a547f1797b2f23c732b2fd436f59c8
    61e1cc2dde0eeb6e6befeb",
    "status": "complete"
}
{
    "destination": "032632259765dd04833258446729d6ef7835c0a7
    b2dcdf3dcc407d09a477a05ab1",
    "payment_hash": "bef9050b75b0ec741892bd213b15365ac707ad14f
    ee9aa6d6e397db93c3acb54",
    "created_at": 1686121157.514,
    "parts": 1,
    "msatoshi": 20000,
    "amount_msat": "20000msat",
    "msatoshi_sent": 21000,
    "amount_sent_msat": "21000msat",
    "payment_preimage": "59da17d9ad54617b37dcda6e6a55a06668dcfa401a
    416b45e6c7ea6c85a288b7",
    "status": "complete"
}
{
    "destination": "032632259765dd04833258446729d6ef7835c0a7b2
    dcdf3dcc407d09a477a05ab1",
    "payment_hash": "889075d9589cba578dc8edec614ce152f1974cc9f
    0ebf34ffdb67d40bdc95b1a",
    "created_at": 1686121161.045,
    "parts": 1,
    "msatoshi": 20000,
    "amount_msat": "20000msat",
    "msatoshi_sent": 21000,
    "amount_sent_msat": "21000msat",
    "payment_preimage": "63eeb3b30e6d6ca6500ffffbdce5530f028181bde4
    89a4479df039ff8d952fc3",
   "status": "complete"
}
{
   "destination": "032632259765dd04833258446729d6ef7835c0a7b2d
   cdf3dcc407d09a477a05ab1",
   "payment_hash": "054d4b18c26afb108b3a0dcf5b05d0134cdbdf60cd
   271bc3b50ff6675bf2c3e1",
   "created_at": 1686121164.667,
   "parts": 1,
   "msatoshi": 20000,
   "amount_msat": "20000msat",
   "msatoshi_sent": 21000,
   "amount_sent_msat": "21000msat",
   "payment_preimage": "a90a521edc6db361c2db556de7e95d75a08cacb385f
   ecf6cfa530e67ca83eb2e",
   "status": "complete"
}
{
   "destination": "032632259765dd04833258446729d6ef7835c0a7b2dc
   df3dcc407d09a477a05ab1",
   "payment_hash": "c490c4d5a2d10542072c2b3aa12f001a1c21096bf9
   937f873d726a765e2efdf9",
   "created_at": 1686121167.813,
   "parts": 1,
   "msatoshi": 20000,
   "amount_msat": "20000msat",
   "msatoshi_sent": 21000,
   "amount_sent_msat": "21000msat",
   "payment_preimage": "aa4153493800dcd4b7ad31f77e09a6bf4b8675426
   9abd4b4a0bb162754bb5072",
   "status": "complete"
}
{
   "destination": "032632259765dd04833258446729d6ef7835c0a7b
   2dcdf3dcc407d09a477a05ab1",
   "payment_hash": "ba3d8c6d3889da651b2d6328f9bbc1ffa57f34331
   4e4c005a5c24dca02e1b836",
   "created_at": 1686121171.118,
   "parts": 1,
   "msatoshi": 20000,
   "amount_msat": "20000msat",
   "msatoshi_sent": 21000,
   "amount_sent_msat": "21000msat",
   "payment_preimage": "a27c33ec47708ba91a4a56cc48d576682d7bdddd7
```

```
    c23917e370ee6bef9e8888c",
    "status": "complete"
}
{
    "destination": "032632259765dd04833258446729d6ef7835c0a7b2
    dcdf3dcc407d09a477a05ab1",
    "payment_hash": "ea102192f8989ecc7606a43613244eb22615a475a
    ef803fb5bf1a98f77bd3a61",
    "created_at": 1686121174.897,
    "parts": 1,
    "msatoshi": 20000,
    "amount_msat": "20000msat",
    "msatoshi_sent": 21000,
    "amount_sent_msat": "21000msat",
    "payment_preimage": "827e65b605db2301069263deaace127d4ad9e1dcb
    4f0875a110fd300f51121c1",
    "status": "complete"
}
{
    "destination": "032632259765dd04833258446729d6ef7835c0a7b
    2dcdf3dcc407d09a477a05ab1",
    "payment_hash": "39c72d4ce697f34219aa9a9ccc731948fd7410598f
    3c44d61aa5f5e07c0a04d1",
    "created_at": 1686121178.657,
    "parts": 1,
    "msatoshi": 20000,
    "amount_msat": "20000msat",
    "msatoshi_sent": 21000,
    "amount_sent_msat": "21000msat",
    "payment_preimage": "c4215e142444cf5ccabdf944044dc8f07908a03f8e
    7e5c4eb6568763cad98f92",
    "status": "complete"
}
{
    "destination": "032632259765dd04833258446729d6ef7835c0a7b
    2dcdf3dcc407d09a477a05ab1",
    "payment_hash": "392abf7328306b672f2c683e7aa73ac76506e7d636
    de970697719edf141b13f6",
    "created_at": 1686121181.868,
    "parts": 1,
    "msatoshi": 20000,
    "amount_msat": "20000msat",
    "msatoshi_sent": 21000,
    "amount_sent_msat": "21000msat",
  "payment_preimage": "9f6b8ee2487daf9ad33a007036cb10d1e7d33e24f
  3f1ef37f6a5648dfb6111e0",
  "status": "complete"
}
{
  "destination": "032632259765dd04833258446729d6ef7835c0a7b
  2dcdf3dcc407d09a477a05ab1",
  "payment_hash": "3580594a46b71aa56cc8d9bf3b86a5ad58f598bb8
  1faff68c19a103840a9a4f3",
  "created_at": 1686121185.425,
  "parts": 1,
  "msatoshi": 20000,
  "amount_msat": "20000msat",
  "msatoshi_sent": 21000,
  "amount_sent_msat": "21000msat",
  "payment_preimage": "4d428a424a5e33e046d73df36ca2c9c06e1969c89
  065c3b668fadf9bb59dd4ac",
  "status": "complete"
}
{
  "destination": "032632259765dd04833258446729d6ef7835c0a7b2
  dcdf3dcc407d09a477a05ab1",
  "payment_hash": "ecdb1d23d94bb37278a77cbc88b7d5aa47164ebb1c
  a7c1083825d71581198233",
  "created_at": 1686121188.680,
  "parts": 1,
  "msatoshi": 20000,
  "amount_msat": "20000msat",
  "msatoshi_sent": 21000,
  "amount_sent_msat": "21000msat",
  "payment_preimage": "2b4f435c65d796ce17da2a52cefaddd95a605a8e90
  f5e61d5d5c52b9b1f46b2c",
  "status": "complete"
}
{
  "destination": "032632259765dd04833258446729d6ef7835c0a7b2
  dcdf3dcc407d09a477a05ab1",
  "payment_hash": "b5050e8b2ec8c9a39290c4b4d1419eac9ea7b29e46
  672e9e75c960b24d0b6b2c",
  "created_at": 1686121191.984,
  "parts": 1,
  "msatoshi": 20000,
  "amount_msat": "20000msat",
  "msatoshi_sent": 21000,
    "amount_sent_msat": "21000msat",
    "payment_preimage": "a1cb313335c46249baddf01e0474b4afd0ab0af966
    33ed2ec09da51c56ed1c04",
    "status": "complete"
}
{
    "destination": "032632259765dd04833258446729d6ef7835c0a7b2
    dcdf3dcc407d09a477a05ab1",
    "payment_hash": "5b770c4f03630ed7b2a14b47934db7fb3d9008ff05
    fe13a9ff6b00e6fb0e6407",
    "created_at": 1686121195.569,
    "parts": 1,
    "msatoshi": 20000,
    "amount_msat": "20000msat",
    "msatoshi_sent": 21000,
    "amount_sent_msat": "21000msat",
    "payment_preimage": "6751e20787c2d3993f220119c040fcc1e6f6ff06c
    336c7edd28fd7bb47ff761b",
    "status": "complete"
}
{
    "destination": "032632259765dd04833258446729d6ef7835c0a7b
    2dcdf3dcc407d09a477a05ab1",
    "payment_hash": "92d11f5b07d765a5dc2682861841ee3d28dd6ea0a
    2c7f375a63862cd15874ef3",
    "created_at": 1686121199.057,
    "parts": 1,
    "msatoshi": 20000,
    "amount_msat": "20000msat",
    "msatoshi_sent": 21000,
    "amount_sent_msat": "21000msat",
    "payment_preimage": "675f2ca7292ea32cb826990b7ce1faaed5395c2338
    6592f3cd444bc3804854c5",
    "status": "complete"
}
{
    "destination": "032632259765dd04833258446729d6ef7835c0a7b2
    dcdf3dcc407d09a477a05ab1",
    "payment_hash": "49cec9b6f595ae7d1eff359a538e1cce9c7e997d8b
    1e1bd28c38bb1e42266b8e",
    "created_at": 1686121202.832,
    "parts": 1,
    "msatoshi": 20000,
    "amount_msat": "20000msat",
    "msatoshi_sent": 21000,
    "amount_sent_msat": "21000msat",
    "payment_preimage": "1575e47aebbf7f46ce1033383a31fdbac3571af5b2
    da4e57062067156fefc78e",
    "status": "complete"
}
{
    "destination": "032632259765dd04833258446729d6ef7835c0a7
    b2dcdf3dcc407d09a477a05ab1",
    "payment_hash": "37a8437296c1b2f28ad781cc63779397f83d0e4d8
    67adab3921afc4a69f6a61c",
    "created_at": 1686121206.747,
    "parts": 1,
    "msatoshi": 20000,
    "amount_msat": "20000msat",
    "msatoshi_sent": 21000,
    "amount_sent_msat": "21000msat",
    "payment_preimage": "1c4a17b8134f65ef24ebde970683dc85238c03cac
    46a7499ea010ee62ed68df7",
    "status": "complete"
}
{
    "destination": "032632259765dd04833258446729d6ef7835c0a7
    b2dcdf3dcc407d09a477a05ab1",
    "payment_hash": "33c6c19b3b395d25c0db1939a982835343278de5
    7e4169336a614b09f8f0529f",
    "created_at": 1686121210.159,
    "parts": 1,
    "msatoshi": 20000,
    "amount_msat": "20000msat",
    "msatoshi_sent": 21000,
    "amount_sent_msat": "21000msat",
    "payment_preimage": "1b4b40567484c404c4cd1392f8a5ab4ee767ba61
    b7e6e6ce52b41beef6756f96",
    "status": "complete"
}
```

Listing A.13 Close transaction of experiment 02

```
{
    "tx": "0200000001283a155bb3bb49e8cdf80ef675df008e1939b
    bb3a0633a64ed871f0e0c7c7ac00100000000ffffffff02c40e00000000000016
    00145ca1305a2cf53fda16ee0126467525a85f2cf829267701000000000022512
    0a2b631637d4eddff8ed05bfc0c4425a0e36ea5b866d0624d8efc26419882bcc
    500000000",
    "txid":
"57f5d1bfc0153f8262755e68d6a032b04b5a6da6defd9b5118d860f288b49dc7",
    "type": "mutual"
}
```

Listing A.14 Close transaction of experiment 03

```
{
    "tx": "0200000001f3f45cd40b2e8b772bea9ccef95f019e25c8
    68d21defcc45a2d9647b0b684e850100000000ffffffff022e27000000000
    0001600142e58dc69e5e60d3438622c975c3e7ea62edb32698be900000000
    00002251203c854c7b7455e84a07539a3eb4b4feb1fadae513d2a0b430fd5
    9d00ef48e94f300000000",
    "txid":
"a586af086a3a8a1029026fe5aefb7181fdfe4d109dc1c5be610062ac9d4de90e",
    "type": "mutual"
}
```

Listing A.15 Transaction on-chain experiment 03

```
{
 "peer_id":
"0270685ca81a8e4d4d01beec5781f4cc924684072ae52c507f8ebe9daf0caaab7b",
 "connected": false,
 "state": "ONCHAIN",
 "short_channel_id": "2436930x12x1",
 "channel_sat": 10030,
 "our_amount_msat": "10030000msat",
 "channel_total_sat": 70000,
 "amount_msat": "70000000msat",
 "funding_txid":
"854e680b7b64d9a245ccef1dd268c8259e015ff9ce9cea2b778b2e0bd45cf4f3",
 "funding_output": 1
}
```

Listing A.16 Output of Node 1 after closing Channel

```
{
 "txid":
"a586af086a3a8a1029026fe5aefb7181fdfe4d109dc1c5be610062ac9d4de90e",
 "output": 0,
 "value": 10030,
  "amount_msat": "10030000msat",
  "scriptpubkey": "00142e58dc69e5e60d3438622c975c3e7ea62edb3269",
  "address": "tb1q9evdc609ucxngwrz9jt4c0n75chdkvnf5mtm2x",
  "status": "confirmed",
  "blockheight": 2436967,
  "reserved": false
}
```