DISTRIBUTED LEDGER TECHNOLOGY BASED MULTI-ENERGY SERVICE PROVISION

A thesis submitted to Cardiff University
in candidature for the degree of
Doctor of Philosophy
by

Andrei Nicolas Manea

Cardiff School of Engineering, Cardiff University
20 September 2024

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to Professor Jianzhong Wu, Dr. Yue Zhou and Dr. Muditha Abeysekera for their steady support, encouragement and guidance during my candidacy. Thank you very much for giving me this opportunity and spending your valuable time to make me a better researcher and person.

I would like to acknowledge the EPSRC for their financial support and Cardiff University for its unwavering support.

My sincere gratitude goes to all members of the CIREGS research group at Cardiff University (both past and present) for the engaging discussions and suggestions regarding my research work.

I am also thankful to my friends, Mr. Karol Podufalski, Mr. Wincenty Dulkowski and Mr. Valery Shchukin for their technical guidance and continuous support through thick and thin.

I would like to thank my mother and my aunt for their love, support and understanding throughout my candidacy.

A special thank you goes to my uncle Daniel Surulescu for his support, love, patience and understanding in what was a very challenging five years. Nothing would have been possible without your support. For this, I will be forever grateful. Thank you.

ABSTRACT

The United Kingdom has pledged to reach Net Zero by 2050, which means removing as much greenhouse gas from the atmosphere as it emits. One of the main challenges associated with this ambitious target is the decarbonisation of the power grid by 2030. Achieving this goal requires a significant transformation of the energy sector, particularly in how energy supply and demand are balanced. Flexibility markets, which facilitate the dynamic adjustment of electricity supply and demand in real-time, play a crucial role in enabling the integration of renewable energy sources and ensuring grid stability. However, the current models for managing these markets face limitations in transparency, efficiency, and cyber security.

This thesis explores blockchain-based models as a novel solution to support the digitalisation of flexibility markets, addressing the before mentioned challenges. Blockchain technology offers a decentralised, secure, and transparent framework that can improve the operation of flexibility markets by enabling more efficient and trustworthy data transactions. Through leveraging smart contracts and distributed ledger technology, blockchain can support the real-time exchange of energy and services between market participants, while also providing a robust platform for verifying and recording transactions. This thesis aims to develop and evaluate blockchain-based models that can improve the efficiency, security, and scalability of flexibility markets. This goal has been addressed through the design and evaluation of three integrated blockchain-based solutions:

a) A distributed ledger technology-based framework and architecture was used to demonstrate the accessibility, transparency and redundancy of deploying the digital environment for facilitation of flexibility services and market in electricity distribution networks.
b) A blockchain-based flexibility settlement mechanism using novel zero knowledge proofs was developed to demonstrate a superior cyber-security method for settling flexibility transactions without disclosing electricity profile information.
c) A blockchain-based financing model using rotating savings and credit associations was developed to demonstrate a sustainable mechanism to finance the purchasing of flexibility assets

Table of Contents

List of figures

List of tables

Nomenclature

Abbreviations

Abbreviation	Definition
API	Application Programable Interface
APR	Annual Percentage Rate
ASIC	Application Specific Integrated Circuits
BFT	Byzantine Fault Tolerance
CMZ	Constraint Management Zone
CPO	Charging Point Owner
CPU	Central Processing Unit
DAPP	Decentralised Application
DCC	Data Central Company
DER	Distributed Energy Resources
DG	Distributed Generator
DLT	Distributed Ledger Technology
DNO	Distribution Network Operator
DPS	Dynamic Purchase Scheme
DSO	Distribution System Operator
DSR	Demand Side Response
DoS	Denial of Service
ENA	Energy Networks Association
ERC	Ethereum Request for Comments
ERC20	Ethereum Request for Comments 20
ERC721	Ethereum Request for Comments 721
ETRM	Energy Trading and Risk Management
EV	Electric Vehicle
EVM	Ethereum Virtual Machine
FP	Flexibility Provider
GB	Great Britain
GO	Group Owner
GPU	Graphical Processing Unit

ITT	Invitation to Tender
IoT	Internet of Things
LV	Low Voltage
MSP	Mobile Service Provider
MVDC	Medium Voltage Direct Current
NESO	National Energy System Operator
NFT	Non-Fungible Token
P2P	Peer to Peer
PBFT	Practical Byzantine Fault Tolerance
PC	Personal Computer
PIN	Periodic Indicative Notice
PQQ	Pre-Qualification Questionnaire
PV	Photo Voltaic
PoA	Proof of Authority
PoET	Proof of Elapsed Time
PoS	Proof of Stake
PoW	Proof of Work
QAP	Quadratic Arithmetic Program
RNG	Random Number Generator
ROSCA	Rotating Savings and Credit Association
SCADA	Supervisory Control and Data Acquisition
SHA	Secure Hash Algorithm
SSC	Stellar Smart Contract
TSO	Transmission System Operator
UK	United Kingdom
V2G	Vehicle to grid
VRF	Verifiable Randomising Function
ZK	Zero Knowledge
ZKP	Zero-Knowledge Proof
zk-SNARK	Zero-Knowledge Succinct Non-Interactive Argument of Knowledge

1 Introduction

1.1 Motivation and objectives

The growing complexity of electricity grids, driven by the integration of renewable energy sources, presents significant challenges for energy management. With the rise of decentralised energy generation, electric vehicles, and flexible demand systems, traditional grid infrastructures must adapt to manage both the increased variability of energy supply and demand. All the above-mentioned challenges result in an increased need for digital infrastructure—one which allows efficient and cyber-safe data exchange among various entities in the sector, and one that promotes a Net Zero transition that is just and fair.

Blockchain technology offers a promising solution for these challenges by providing secure, transparent, and decentralised solutions that enable new models for energy markets, asset financing, and energy community management. As electricity systems evolve to accommodate new energy actors and technologies, the role of decentralised digital environments becomes more crucial. This thesis is motivated by the potential of blockchain and distributed ledger technologies (DLTs) to address these systemic challenges and facilitate the development of flexibility markets at the distribution level.

The main objective of the thesis is to explore if decentralised ledger technologies can provide a self-sufficient digital infrastructure for the deployment of flexibility services in local, distribution network markets. This involves assessing the technical and operational feasibility of blockchain as a foundation for decentralised energy services, and whether such an approach can deliver resilience, efficiency, and fairness in grid operations.

1.2 Research Questions

In support of the thesis objective, the central research question being addressed is: Can blockchain technology provide an efficient digital environment where market

participants can conduct all their market activities in a decentralised manner, thus promoting a transition that is resilient and fair? To explore this question, each of the following technical chapters investigates a focused set of sub-questions relevant to their specific contributions.

The first technical chapter examines the viability of blockchain for enabling decentralised local flexibility markets, and compares its performance with alternative ledger technologies:

- Can blockchain offer a digital environment for the deployment of a flexibility market platform, where all the relevant market characteristics can be implemented in a decentralised manner, for the local flexibility markets in GB?
- What are the benefits and downsides of deploying this framework on a blockchain, when compared to deploying this framework on a hash graph in terms of latency, throughput, and transactions per second?

The second chapter introduces a blockchain-based settlement mechanism using Zero-Knowledge Proofs (ZKPs) and investigates its potential for secure and privacy-preserving flexibility verification:

- In settling a flexibility transaction, can the current baselining methodologies be transposed to a zero-knowledge algorithm which allows the DNO to receive the same level of information, but with no data exchanges?
- Is the MINA blockchain developed enough to support such an algorithm, thus exploiting MINA's advantages of being a lightweight blockchain?

The third technical chapter addresses community-based financing through decentralised ROSCAs (Rotating Savings and Credit Association) and evaluates its comparative financial performance:

- Can the characteristics and architecture of a ROSCA be transposed on a blockchain, thus allowing for the development of a blockchain-based ROSCA algorithm which promotes the adoption of flexibility assets in a microgrid?
- How does the ROSCA compare, in net present value, with the more traditional ways of purchasing a flexibility asset, in particular conducting monthly savings and using a financial borrowing tool (such as a car lease for an EV)?

1.3 Contributions and publications

This thesis contributes a set of novel digital frameworks for decentralised energy service deployment. These include a blockchain-based architecture for local flexibility markets, a privacy-preserving settlement mechanism based on ZKPs, and a decentralised financial model using smart contract-enabled ROSCAs. These frameworks aim to enhance the transparency, security, and accessibility of energy systems, with particular focus on the local distribution network level.

The findings and methodologies developed in this thesis have been disseminated through the following publications:

- Y. Zhou, N. Manea et al., "Application of Distributed Ledger Technology in Distribution Networks," in Proceedings of the IEEE, vol. 110, no. 12, pp. 1963–1975, Dec. 2022, DOI: 10.1109/JPROC.2022.3181528;
- N. Manea, Akoury-Shima, M., Bhandari, V. and Stammler, J. L., 2024. Decarbonisation demands decentralisation: how blockchain is impacting the energy sector. IET Conference Proceedings, 2024(5), pp. 1048–1052, DOI:10.1049/icp.2024.1953.
- N. Manea (2022) "Blockchain applications for flexibility services," presented at the SUPERGEN 2022 Conference, Cardiff, 5 September 2022.

1.4 Structure of the thesis

The thesis is structured such that each chapter builds upon the previous one. This chapter presents the motivation behind the study, an introduction, and the research questions for all of the technical contributions. The literature review chapter informs the reader on distributed ledger technologies, explaining a range of technical details and presenting a state-of-the-art review of blockchain applications in the energy sector. The chapter ends with a summary identifying the gaps in the literature. This summary builds on the initial motivation for the next three chapters, where each is a technical contribution, and they are introduced below.

The final chapter presents the conclusion of the thesis and a reflection on the limitations of the work alongside a list of relevant directions the research can be progressed. The first technical chapter, titled A DLT-Based Framework for Flexibility Services Market Facilitation, proposes a novel framework based on DLT to manage flexibility services in a decentralised manner. By automating the processes through smart contracts, this framework increases transparency, reduces operational inefficiencies, and ensures a secure transaction environment for market participants. The framework focuses on ensuring that flexibility service providers (such as households with DERs) and grid operators can interact seamlessly without requiring a central authority. It leverages blockchain's immutable record-keeping and consensus mechanisms to track flexibility contributions and energy transactions, making it more resilient to fraud and manipulation.

The second technical chapter, Blockchain-Based Flexibility Settlement with ZKPs, introduces a blockchain-based settlement mechanism using ZKPs . This approach ensures that flexibility transactions can be verified and settled securely without exposing private data, thus maintaining both the privacy of the service provider and the integrity of the market. The chapter explains how ZKPs can be implemented within a decentralised flexibility market framework, allowing for the validation of energy adjustments without disclosing specific details of participants' energy usage. This solution enhances the confidence of flexibility market participants by allowing them to engage in the market without fear of their data being compromised. Furthermore, it improves the scalability of the blockchain system by reducing the amount of data that needs to be processed and stored on-chain, which is critical for large-scale energy market implementations.

The third technical chapter, Decentralised ROSCAs for Financing Flexibility Assets, details how smart contracts can automate the operation of ROSCAs, ensuring that participants' contributions are securely managed, and the distribution of assets follows predefined rules. By decentralising the financial process, this approach reduces the reliance on traditional financial institutions and opens up opportunities for a wider range of participants to invest in energy assets. Additionally, the transparency of blockchain technology ensures that all transactions are visible and

verifiable, which builds trust among participants and eliminates the risk of fraud or mismanagement.

The final chapter presents the overall conclusion of the thesis, summarising the key findings and contributions. It also provides a reflection on the limitations of the work and outlines a set of potential directions for future research, particularly focusing on the scalability, interoperability, and regulatory alignment of blockchain-based energy systems.

2 A review of blockchain applications in the energy sector

2.1 Introduction

The literature review for this thesis begins by exploring the foundational concepts and evolution of DLT, particularly in the context of its potential to transform the energy sector. The review examines the technical features and key innovations of DLT, and how these have been leveraged in various industries. In particular, it focuses on the applicability of DLT to flexibility markets within the energy sector, a critical area for achieving the United Kingdom's (UK) ambitious decarbonisation goals. By analysing existing research, pilot projects, and real-world implementations, the literature review identifies the benefits, challenges, and gaps in the current understanding of DLT's role in energy market digitalisation. This provides the necessary context to assess the viability of blockchain-based models for supporting the digitalisation of flexibility markets.

DLT is considered a promising innovation that could significantly impact the delivery of public and private services, improving productivity across various applications, as noted in a report by the GB Government Chief Scientific Adviser ("Distributed Ledger Technology: beyond block chain," 2021). Since its debut in 2009, DLT has gained increasing attention and has developed quickly.

DLT is essentially a type of distributed database that is collectively maintained by the nodes in a network, without the need for a trusted centralised authority. The data, or "ledger," exists in multiple copies, with each eligible node holding one. Data is

updated and kept consistent across the network through a consensus mechanism. One of the most common methods is “Proof of Work” (PoW), where updates are made by the winner of a computational race in which all eligible nodes participate in, with those having more computing power being more likely to succeed. In blockchain, a well-known form of DLT, data is stored in blocks that are linked in chronological order, with each block containing a reference to the previous one, making it difficult to alter the data.

An analogy helps to illustrate DLT (illustrated in Figure 2:1) can be the following: imagine a group of people whose assets and transactions are recorded in ledgers. In a traditional centralised database, one person maintains the ledger for everyone, which allows the possibility of errors or intentional changes. In contrast, with DLT, everyone keeps their own copy of the ledger, and changes can only be made following a set of agreed-upon rules, ensuring that no single person can easily alter the ledger.

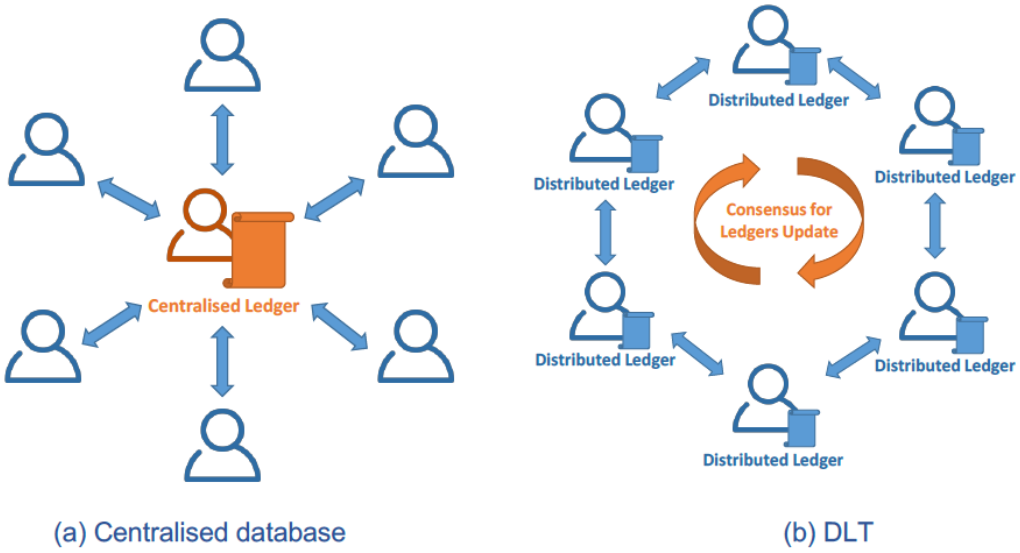


Figure 2:1 Centralised database vs. DLT

The implications of DLT extend beyond merely storing data in distributed databases. DLT enhances trust in digital processes, enabling the use of smart contracts, which are self-enforcing agreements. By fostering a new level of trust, DLT has the potential to transform a variety of private and public services. This could lead to significant reforms in areas such as financial markets, supply chains, consumer and business-to-business services, and publicly held registers.

2.1.1 What are the key features of DLT?

There are numerous variants of DLT, each with different implementations across its various components. However, the key features of DLT can generally be discussed in several areas, including public verifiability, transparency, privacy, integrity, and redundancy, as noted by Wüst and Gervais (Wüst and Gervais, 2017). When compared to centralised databases, the features of DLT are as follows:

- Public verifiability: DLT allows any participant to verify the correctness of the data, a feature not typically found in centralised databases.
- Transparency: DLT enables participants to observe both the data and the process of updating it. Although not every participant may access every piece of information, DLT generally offers higher transparency compared to centralised databases, which usually have lower transparency.
- Privacy: While privacy can conflict with public verifiability and transparency, DLT can still maintain a certain level of privacy. For example, DLT can make data transparent and publicly verifiable while keeping participants anonymous. In contrast, centralised databases can more easily guarantee privacy since they do not need to provide public verifiability and transparency.
- Integrity: DLT offers high data integrity because any participant can verify the data. In centralised databases, data integrity largely depends on the centralised authority.
- Redundancy: DLT inherently has high redundancy because data copies are distributed across the network's nodes. Centralised databases can achieve redundancy by copying data across different hardware.

Beyond these high-level features, specific characteristics of DLT are compared with those of centralised databases and summarised in the table In Table 2:1. It is important to note that DLT can be categorised into permissionless and permissioned types. In permissionless DLT, anyone can participate, whereas in permissioned DLT, only authorised participants (appointed by a centralised entity) are allowed to join.

Technical Features	Permissionless DLT	Permissioned DLT	Centralised Database
Throughput	Low	High	Very high
Latency	Slow	Medium	Fast
Number of participants allowed to access data	High	High	High
Number of participants allowed to update data	High	Low	High
Number of untrusted participants allowed to update data	High	Low	None
Consensus mechanism	Mainly "Proof of Work", Some "Proof of Stake"	Byzantine Fault Tolerance	None
Centrally managed	No	Yes	Yes

Table 2:1 Technical features of DLT

2.1.2 How was DLT developed?

In 2009, Bitcoin, the virtual cryptocurrency, was introduced, marking what is generally recognised as the birth of DLT. Although the terms "DLT" or "blockchain" were not explicitly used at that time, DLT is the foundational technology behind Bitcoin. According to Swan (Swan, 2015), the development of DLT can be categorised into three stages: DLT 1.0, DLT 2.0, and DLT 3.0, as illustrated in Figure 2:2.

The foundation of DLT is built on several key technologies, including peer-to-peer networking, cryptography, distributed databases, and digital currency. In 2008, the Bitcoin white paper was published, and in 2009, the Bitcoin system was launched, marking the beginning of DLT 1.0. DLT 1.0 is characterised by the emergence of cryptocurrencies like Bitcoin. It is a decentralised, transparent ledger that records transactions—a database shared by all network nodes, updated by miners, monitored by everyone, and owned or controlled by no single entity.

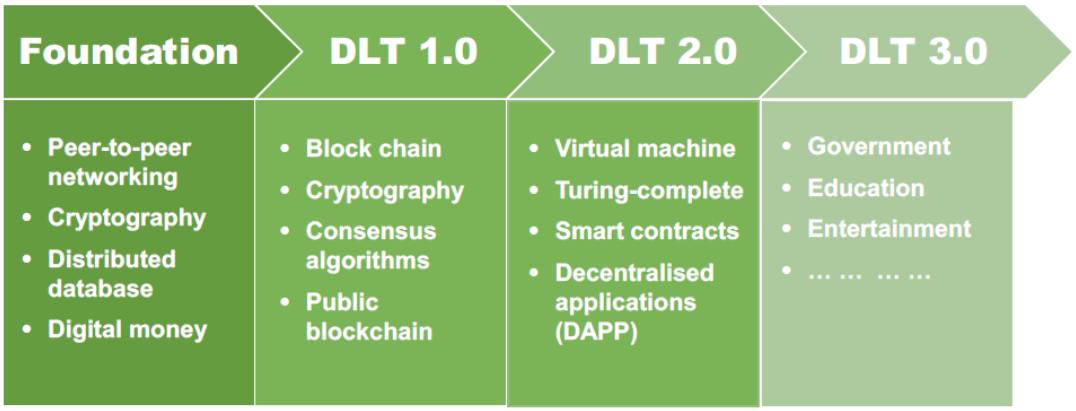


Figure 2:2 DLT evolution (Swan, 2015)

Around 2014, DLT 1.0 began to transition into DLT 2.0, which introduced the capability to register, confirm, and transfer various types of contracts and property. DLT 2.0 is defined by the development of Turing-complete virtual machines, smart contracts, and decentralised applications (DAPPs).

DLT 3.0 extends DLT's applications beyond currency, economics, and markets, focusing on broader uses rather than purely technical advancements. While DLT 2.0 represents the most advanced technology to date, its applications in new fields, such as electrical systems, are considered part of the DLT 3.0 stage.

2.1.3 Do I need DLT?

DLT has garnered significant attention in recent years, with companies across various industries exploring its potential to enhance their operations. The number of start-ups basing their business models on DLT is rapidly growing, and numerous research and trial projects involving DLT are currently underway. However, it is crucial to carefully assess whether the promising features of DLT will truly deliver benefits, as well as to consider the associated costs and risks.

Peck cautioned against hastily rebuilding the entire digital ecosystem on DLT(Peck, 2017). He provided a flow chart designed to help determine whether DLT is the right choice, based on a series of questions. This chart emphasises key factors such as the need for data updates, information security, redundancy, privacy, authority management, and the level of trust among participants. It also encourages considering whether traditional database technologies might meet these needs at a

lower cost. Wüst and Gervais offered a similar discussion and flow chart. It is important to note that the flow chart in Figure 2:3 highlights only some key factors and serves as a guide for decision-making.

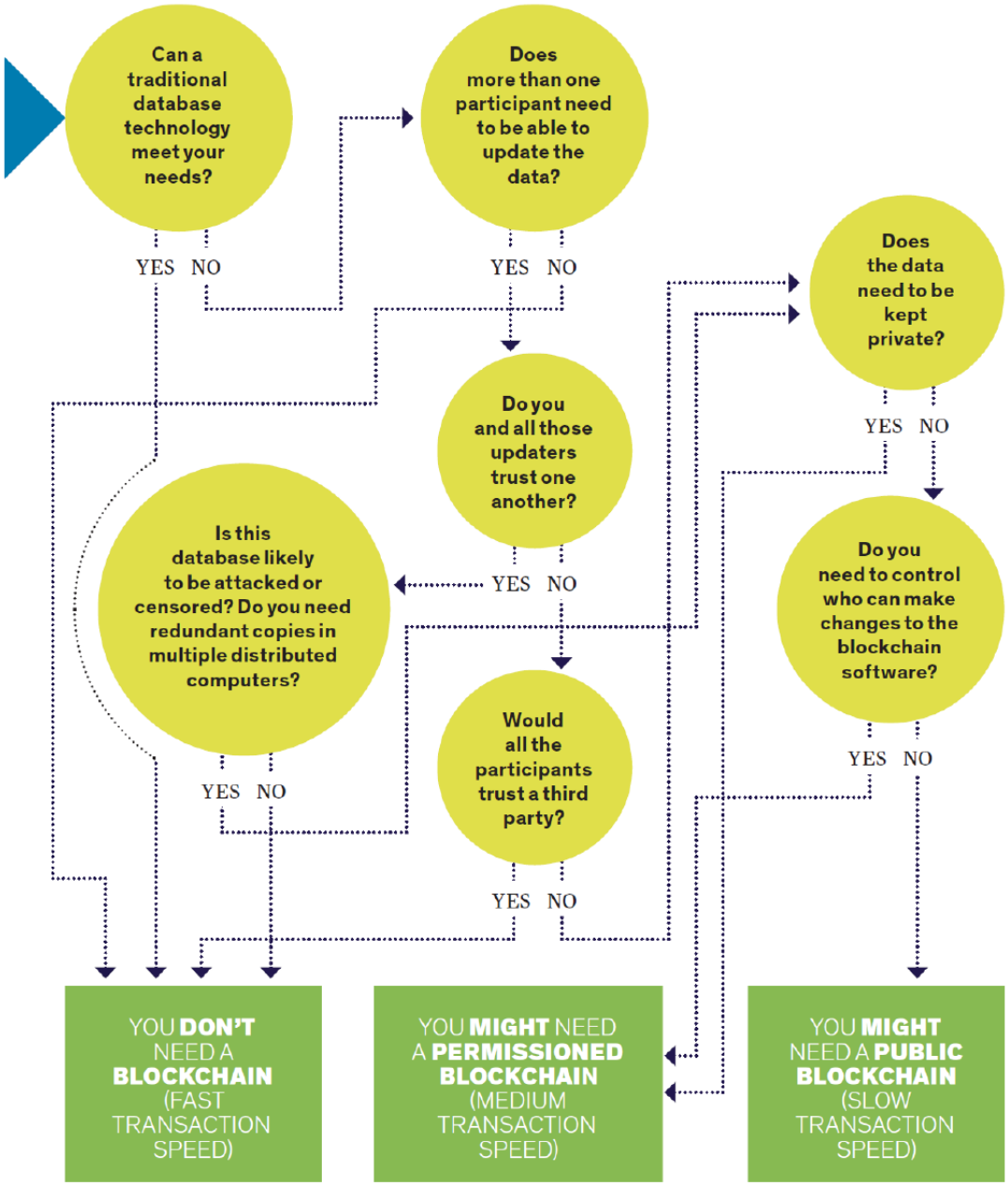


Figure 2:3 Peck's Flow (Dütsch and Steinecke, 2017)

From the perspective that the motive to use DLT is to establish trust, the question of whether and when DLT is needed can be answered from another angle, which assesses the need of DLT through the evaluation of risk and cost of trust breach.

2.1.4 How could I use DLT?

DLT could be used in a wide range of areas, and there are a large number of applications, trials and research activities being conducted or planned. Some example areas and use cases are given in Figure 2:4 (Dütsch and Steinecke, 2017).

	Industry	Use cases
	Energy, utilities & mining	• Smart utility metering system • Decentralised energy data platform
	Entertainment & media	• Control of ownership rights of digital media • Disintermediation of record labels
	Financial services	• International P2P transactions • Anti-money laundering
	Government & public services	• Land ownership records • Tamper-proof voting records • Digital identity of citizens
	Healthcare	• Storage of healthcare records • Population health and clinical studies
	Hospitality & leisure	• Loyalty programmes
	Insurance	• Peer-to-peer flight insurance policies • Micro-insurance
	Transportation & logistics (freight transport)	• Trade documentation (e.g. Bill of Lading) • Trade finance • Supply chain transparency
	Transportation & logistics (aviation)	• Distribution of tickets and ancillary services • Loyalty programmes (cf. H&L) • Passenger identity management

Figure 2:4 Examples of DLT Applications

2.2 An overview of DLT

2.2.1 Why Distributed Ledgers?

A key motivation for adopting DLT is to enhance trustworthiness in digital processes, particularly in the accuracy of stored information and the integrity of the rules governing changes to that information. DLT achieves this by replacing third parties—

who can be vulnerable to manipulation, misconduct, or errors—with a distributed system that operates across a network of computers, all adhering to a predefined set of consensus rules. This approach is particularly useful in applications involving the transfer of value or information based on pre-agreed conditions. DLT enables self-enforcing agreements, known as smart contracts, which are characterised by their trustworthiness, replicability, and verifiability. These contracts facilitate the transfer of value based on digital inputs and provide tamper-resistant information storage, making the technology potentially valuable for Distributed Network Operator (DNO) businesses.

The concept of trust within DLT can be further detailed, as shown in Figure 2:5. For a DLT implementation to be successful, it must be reliable across each of the domains listed in the table. In marketing literature, DLT is sometimes described as "trust less," suggesting that users do not need to invest effort in determining which third parties to trust. Another perspective is that DLT users are reshaping the landscape of trust by shifting focus on who and what they trust, with the potential benefit of achieving high levels of trust and the automatic enforcement of contracts tied to digital events.

Conceptual agreements	Business processes	Technical profiles
Operational agreements	Certificate policies	Legal subscribers
	Certification practice statements	Small community organizations
Governance structures	Business agreements	Registration practices
Assurance metrics	Determining the state of a digital credential at a point in time	Trust-list management
Credential system	Privacy frameworks	Cryptography
Certificate policies	Registration authority	CA hierarchies
		Trust stores
Certificate practices	Directories	Revocation
Cryptography	Key management	
Key escrow	Validation practices	Software engineering

Figure 2:5 Concept of trust in DLT

There is a wide array of technologies encompassed under the umbrella of DLT. The challenge for DNOs lies in understanding how to effectively evaluate DLT and identify potential use cases that could benefit their current and future operations. The

initial hypothesis suggests that DLT applications are most relevant in scenarios where the risk of a trust breach (a deviation from an expected outcome due to malfunction or malfeasance) is relatively high, or where the cost of such a breach is significant, as illustrated in the in Figure 2:6. It is important to ensure that the application can be reliably translated into the digital domain. A key challenge in assessing prospective DLT applications is finding methods to evaluate the likelihood of a trust breach and the potential costs associated with it.

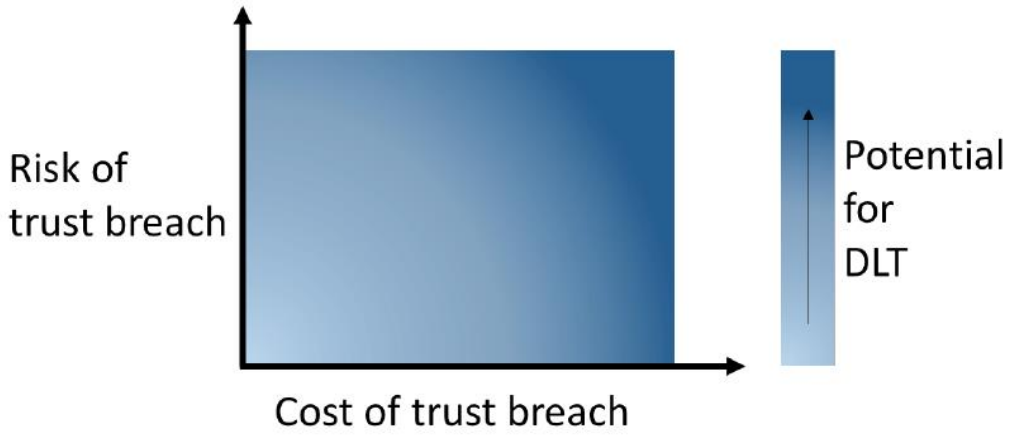


Figure 2:6 The potential of DLT

2.2.2 Understanding DLT

DLT refers to systems that enable multiple parties to reach a consensus on the state (structure and contents) of a shared digital object, such as a ledger, across a distributed network of computers. The ledger's state is updated through specially formatted instructions known as transactions. These transactions are typically grouped into distinct blocks, which are then agreed upon across the network. Each block includes a reference to the previously agreed-upon block, forming a chain of blocks commonly known as a blockchain. This is why the term "Blockchain Technology" is often used interchangeably with DLT, although other configurations also exist, as illustrated in Figure 2:7.

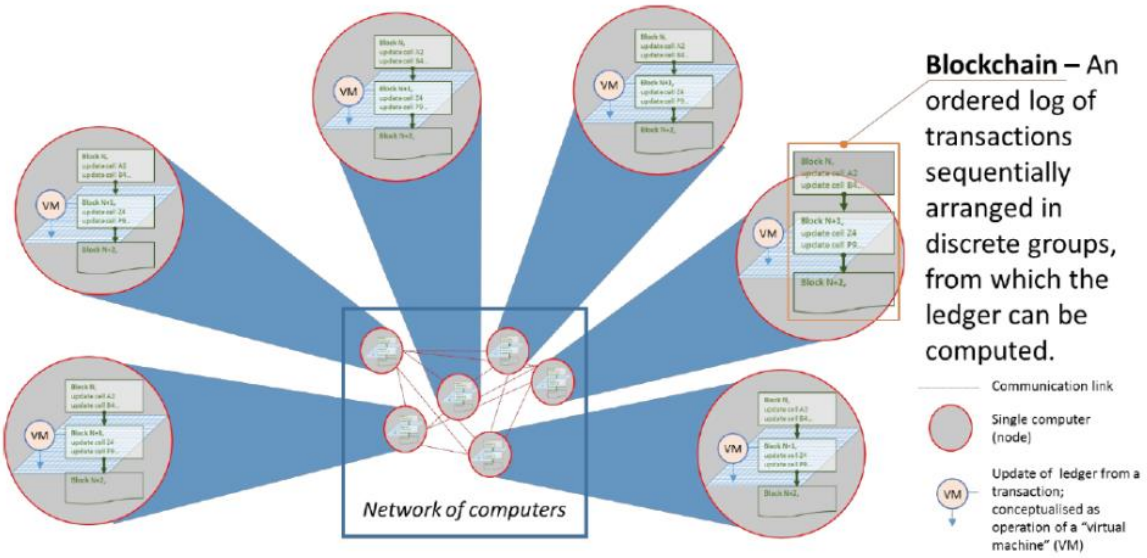


Figure 2:7 Blockchain illustration

DLT systems are classified as either Permissioned or Permissionless, depending on the rules governing participation in the network. Permissioned systems restrict access to a select group of participants, such as the computers belonging to specific companies. In contrast, Permissionless (or Public) systems allow any computer to join the network. Private systems are those operated on a network of computers owned by a single entity, typically used for testing purposes. Sometimes, the term "private" is also used to describe Permissioned systems.

The computers that make up, participate in, and validate the state of a DLT network are called "nodes." In contrast, computers that connect to the network to gather information or submit transactions, but do not participate in validating the state, are known as "clients" or "light clients." Light clients are common in public DLT systems where individual computers may have limited computational resources. In the context of energy systems, there are proposals to use the computers within smart meters as light clients.

2.2.3 Smart Contract Virtual Machines

When a transaction is received, each node in the DLT network computes an updated state of the shared digital object. This process can be conceptualised as being carried out by a virtual machine. The instruction set and operational rules of this

virtual machine can be defined to include protocols for transferring value between parties. This capability facilitates the creation of smart contracts, which are "self-enforcing agreements in the form of executable programs" (Bracciali et al., 2018). The concept is illustrated in Figure 2:8.

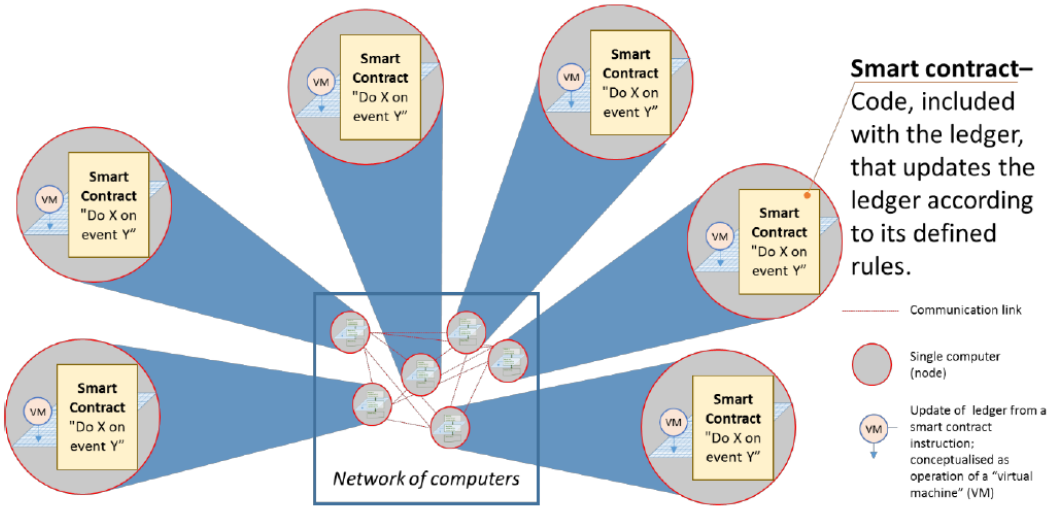


Figure 2:8 Smart contracts concept

The potential of DLT for DNOs largely stems from the use of smart contracts. The shared virtual computer can execute operations on the shared ledger, typically by incorporating specific instructions within transactions. This setup enables the exchange of value based on the occurrence of predefined events. The code that defines these rules (e.g., if X happens, pay Y to Z) is referred to as a smart contract. In public permissionless DLT systems, this allows for self-enforcing value transfers. In permissioned systems, the process may involve the generation of promissory notes, which could then be enforceable through legal means. The concept of smart contracts was originally proposed by Nick Szabo (Christidis and Devetsikiotis, 2016a; Szabo, 2022).

One commonly used smart contract virtual machine is the Ethereum Virtual Machine (EVM) (Wood, 2014). The EVM is employed in both public permissionless systems (such as Ethereum and Ethereum Classic) and permissioned systems (like Hyperledger Fabric). It defines a set of operations in an assembly language. Several higher-level programming languages and compilers, such as Solidity, Vyper, and LLL, have been developed to create smart contracts. Currently, there is ongoing

development of a new smart contract virtual machine based on WebAssembly, an existing instruction format used for web applications.

For clarity, the illustrations in Figure 2:8 depict the shared object as a ledger, with transactional updates represented as "cells" similar to a spreadsheet. In practice, the shared object could take various other forms. Additionally, while transactions are typically organised into a reference chain of blocks, known as a blockchain, there are more complex alternative structures that have been proposed, such as Hedera Hashgraph or Directed Acyclic Graphs. However, the core principle remains the same: storing transactions in a manner that is resistant to tampering and allows for verification.

2.2.4 Cryptography

A cryptographic hash function translates any digital object of arbitrary length into a fixed length "fingerprint." These functions are one-way, meaning that while it is easy to generate the fingerprint from the input, it is extremely difficult to reverse the process and determine the original input from the fingerprint. If the output of a hash function is Y bits, there are 2^Y possible outputs. The exact position of the output within this 2^Y range is highly unpredictable.

Figure 2:9 illustrates the concept of a cryptographic hash function, with Y set to 16 bits for clarity. In practice, Y is typically 256 or 512 bits, leading to an "output space" with 2^{256} or 2^{512} possible combinations. This vast output space makes the likelihood of a "hash collision," where two different inputs produce the same output, very low. An example of a hash function is SHA3 (Secure Hash Algorithm) (“Public key cryptography, public domain images’,” 2021) is presented in Figure 2:9.

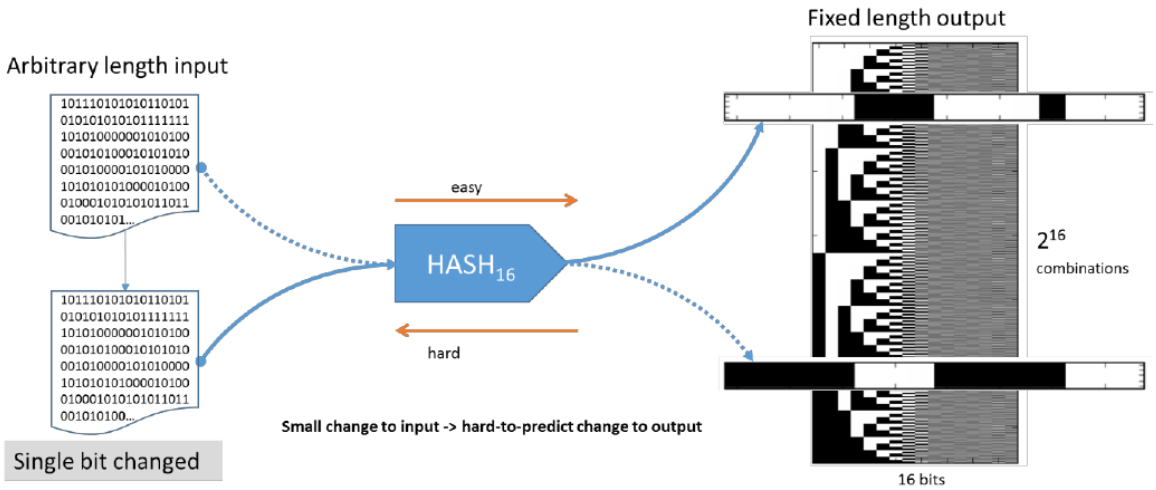


Figure 2:9 Cryptographic Hash Function

Digital information can be broken down into smaller components, each referenced by its hash value (the fingerprint produced by a hash function). A Merkle tree, named after Ralph Merkle, organises this information into a tree-like structure where parent nodes are formed by hashing the values of their child nodes, as shown in Figure 2:10. The topmost node, known as the root or top hash, is the single parent of the entire tree. If any information changes at a distant branch, the hashes along the path to the top hash—including the top hash itself—will also change. This structure provides an efficient way to verify that a large dataset has remained unchanged. In the context of DLT, a Merkle tree can be used to store hashes of transactions, providing proof of which transactions were included and validating their integrity.

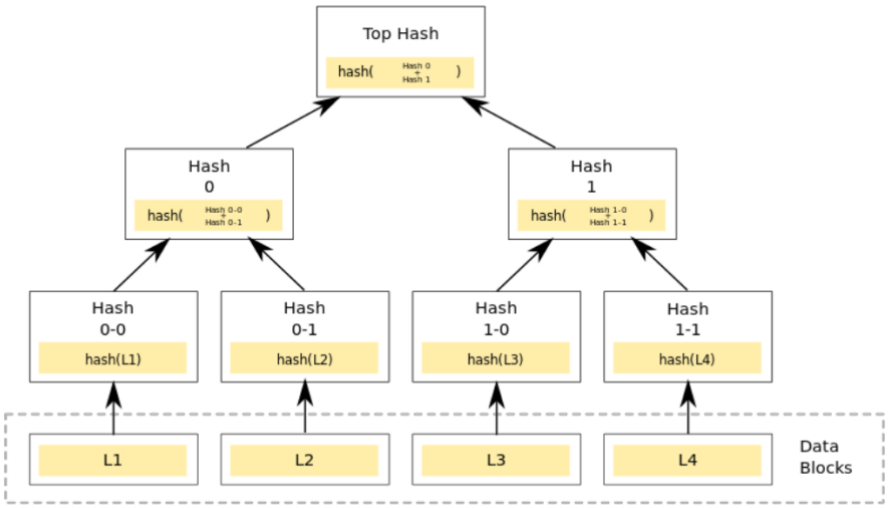


Figure 2:10 A Merkle Tree (hash-function)

Public key cryptography is a category of cryptographic functions where pairs of mathematically related numbers are generated, forming a public key (which can be shared with anyone) and a private key. These keys are related in such a way that it is nearly impossible, with traditional computing methods, to determine the private key from the public key. The relationship between these pairs allows information encrypted with a public key to be decrypted only with its corresponding private key, ensuring that the message remains confidential between the sender and the receiver.

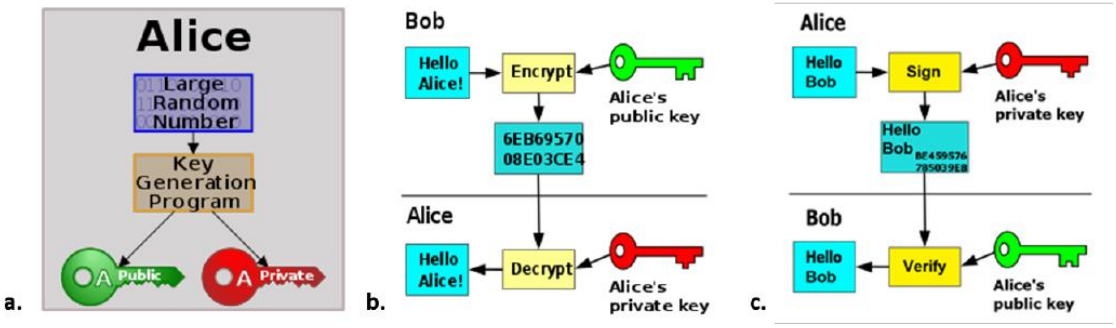


Figure 2:11 Public key illustration

Another application of public key cryptography is in digital signatures. Typically, the contents of a message are passed through a cryptographic hash function, and the resulting "hash" is then encrypted by the sender using their private key. The recipient can verify that the message originated from the sender by hashing the message themselves and comparing it to the decrypted signature (using the sender's public key). This process is illustrated in Figure 2:11.

A crucial step in public key cryptography is the generation of key pairs. This is accomplished through mathematically complex problems (e.g., based on integer factorisation, discrete logarithms, and elliptic curves). The generation process typically requires a large random number as input, making it essential that this number is truly random and that the key generation process remains secure to ensure the overall security of the system.

Users of DLT rely on the secure generation and storage of key pairs, particularly the private key. The hardware and software used for this purpose are critical components of the trust framework, which must be carefully evaluated when

considering the viability of using DLT for a specific application. Cryptographic hash functions and public key cryptography are often used in DLT as the foundation for a currency, where the ledger tracks account balances and transactions facilitate the transfer of value between accounts. Typically, account numbers are derived from a user's public key. The originator of a transaction is authenticated using a digital signature, enabling the transfer of value. In some consensus protocols, the currency acts as an incentive, facilitating agreement on the ledger's state. The growth and development of DLT-enabled currency, commonly known as cryptocurrency, is well documented (Peters et al., 2015).

2.2.5 Consensus mechanism

To ensure agreement on the state (structure and contents) of the shared digital object, such as a ledger, DLT systems employ rulesets designed to maintain consensus across the network of computers. A critical challenge in preventing such a breakdown lies in designing the rules so that no single party or subset of parties can corrupt the ledger. Consensus protocols, or consensus mechanisms, establish these rules to maintain a shared version of the object's state among all parties involved. These protocols integrate knowledge from various fields, including game theory, cryptography, economics, computer science, and psychology. A detailed comparison of these protocols was conducted by Bano et al. (Bano, 2021). An illustration of consensus breakdown is presented in Figure 2:12.

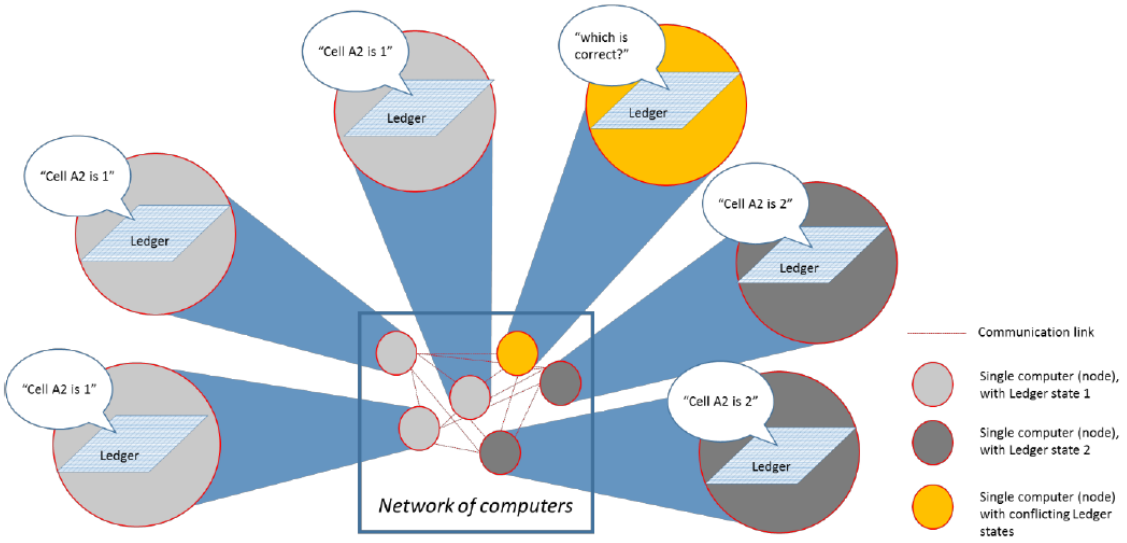


Figure 2:12 Illustration of a breakdown in consensus in a DLT system

In systems that require consensus across the network, a Byzantine Fault is a failure that presents different symptoms to different observers. Byzantine Fault Tolerance (BFT) refers to the ability of a network of computers to continue operating effectively despite such faults. Byzantine failure occurs when a system service is lost due to a Byzantine fault in systems requiring consensus. The concept of BFT is foundational in the development of distributed systems, including DLT.

The term Byzantine originates from the Byzantine Generals' Problem, introduced by Lamport et al. in the early 1980s (Lamport et al., 1982). The problem is a metaphor for the challenge in a computer network where some computers might send false or incomplete information. An early implementation of a BFT algorithm is the Practical Byzantine Fault Tolerant (PBFT) algorithm (Castro and Liskov, 2002), which is now a class of algorithms used in permissioned DLT systems like Hyperledger Fabric (Cachin, 2016).

PoW is a consensus mechanism where computers in the network are incentivised (often with a built-in currency) to calculate a valid future state of the shared object. This typically involves collecting transactions that define valid changes to the object. The computers then engage in a computational race, combining the state change information with arbitrary numbers until a number within a specific range is output. Once a valid block is found, it is broadcast to the network, and the process repeats for the next state. This race, involving cryptographic hash functions, is the essence of PoW systems.

In PoW, the network's hash rate—the rate at which cryptographic hash functions are performed—affects the difficulty of finding a valid block. The system adjusts the threshold to maintain a consistent time between blocks, regardless of the network's hash rate. Bitcoin, Ethereum, and Ethereum Classic are examples of PoW systems. Bitcoin, in particular, has operated since 2008 without known compromise, though critics argue that the energy consumption required for mining is excessive, leading to the development of alternatives like Proof of Stake (PoS).

In PoS, consensus is achieved by having each participant place a deposit, or stake, against the validity of a future block. Computers that bet on an incorrect state risk

losing their stake. Like PoW, PoS systems are often paired with an underlying cryptocurrency. The design and implementation of PoS protocols are active areas of research and development, with notable implementations including Casper (Ethereum) (Buterin and Griffith, 2017), Ouroboros (Cardano) (Kiayias et al., 2017), and Hedera Hashgraph (Baird, 2021; Baird et al., 2021).

Delegated Proof of Stake (DPoS) is a variant of PoS where participants place their deposits against elected delegates—computers responsible for validating future blocks. Examples of DPoS implementations include EOS and Lisk (“Lisk webpage’,” 2021).

Proof of Authority (PoA) is a consensus mechanism where a subset of the network's computers, known as authorities, are designated to validate future blocks. This approach can increase the rate of changes to the shared digital object. An example is the POA.network (Authority LLC, 2021), where validators are publicly declared individuals whose details, including physical addresses, can be verified. The Energy Web Foundation (Foundation, 2018) highlights potential risks with PoA, such as centralisation and increased reliance on timestamp accuracy, and proposes mitigations like a transparent validator selection process.

In Proof of Weight, a computer's ability to validate future blocks is weighted based on its contribution to the network, such as the amount of file storage provided. Examples of systems using Proof of Weight include Algorand (Gilad et al., 2017), (Gilad et al., 2017), Filecoin (IPFS) (“Filecoin webpage’,” 2021), and Chia (“Chia platform webpage’,” 2021).

Proof of Elapsed Time (PoET) is a consensus algorithm developed by Intel that relies on Intel's SGX technology—Trusted Execution Environments designed to prevent tampering. In PoET, each participant waits a randomly generated length of time before validating the next block, with trust in the system depending on the security of the Trusted Execution Environment (Rilee, 2021).

Beyond the computer-based consensus protocols, consensus among the parties involved in the DLT system is necessary for the overall operation. This includes

agreeing on the consensus protocols themselves, the computer network protocols, the nature of the shared digital object, and the rules for updating it. From a DNO perspective, this agreement involves selecting a particular platform or designing a new one. The consensus can be broadly categorised into social (agreeing on the overall system's description) and computer-based (networking and consensus protocols).

2.2.6 Limitations and privacy

A common criticism of public permissionless DLT is the limited transaction rate, or the speed at which the state of the shared digital object can be updated. This limitation arises partly because it takes time for state update information to propagate across a distributed network and because consensus protocols require time to operate. As a result, permissioned DLT systems generally achieve higher transaction rates, and centralised databases can achieve even faster transaction rates. The challenge of creating scalable networks is a major focus of ongoing research and development. Some of the proposed solutions to address these scaling issues include:

- Casper: A consensus algorithm being developed for the Ethereum network.
- Cardano: A platform under development that uses PoS (Cardano, 2021; Kiayias et al., 2017)
- Hedera Hashgraph: Aims to reduce reliance on the propagation of all transactions to all nodes by organising transaction blocks into a graph structure (Baird, 2021; Baird et al., 2021).
- Lightning Network: A second-layer solution built on top of the Bitcoin blockchain to increase transaction speed (Poon and Dryja, 2016).
- Plasma Network: A proposed second-layer solution for the Ethereum blockchain to improve scalability(Poon and Buterin, 2018).

Achieving privacy in DLT is inherently challenging, as transactions are broadcast across a network of computers. However, techniques such as zero-knowledge proofs offer a way to verify information without exposing it. These techniques are being adapted for use in DLT, although they are still in relatively early stages of development in this context. Examples of privacy-focused public DLT projects

include Zcash (Peck, 2016) and Monero (Lee and Miller, 2018), both of which employ advanced cryptographic methods to enhance transaction privacy (Henry et al., 2018). Active research in this area continues, with studies like that of Aitzhan and Svetinovic exploring security and privacy in decentralised energy trading and implementing proof-of-concept solutions (Aitzhan and Svetinovic, 2018).

2.2.7 Existing Blockchain Platforms

Bitcoin, introduced in 2008, is an electronic currency that marked the first widespread use of PoW as a consensus protocol. As a permissionless system that has been in operation for 10 years as of 2018, Bitcoin's network is maintained by developers (primarily the Bitcoin Core team) and social influencers. Despite being a foundational platform in the DLT space, Bitcoin does not feature a smart contract virtual machine. However, the Lightning Network, a second-layer micropayment system rooted in Bitcoin, operates in a permissionless environment and introduces bespoke consensus mechanisms that allow for smart contract functionality. (Nakamoto, 2008; Poon and Dryja, 2016).

Cardano is a platform that aims to support smart contracts, decentralised applications, side chains, multi-party computation, and metadata management. Launched as a permissionless system, Cardano's consensus is maintained through the PoS protocol known as Ouroboros. The platform, which had been in operation for one year by 2018, is heavily influenced by IOHK (Input Output Hong Kong) and social influencers, and it includes a smart contract virtual machine (Cardano, 2021). Chia is a pre-launch digital currency and blockchain that relies on proofs of space and proofs of time instead of the traditional PoW. Although Chia had not launched as of 2018, it is designed as a permissionless system with a bespoke consensus mechanism, heavily influenced by the Chia company, its developers, and social influencers. The details of its smart contract capabilities were not disclosed at that time (“Chia platform webpage’,” 2021).

R3's Corda, a permissioned DLT platform, is the result of a collaborative effort involving R3 and industry partners. Corda allows for multiple consensus algorithms and is influenced by the R3 company and its partners. Corda supports smart contracts via the EVM (“R3 platform website’,” 2021).

Ethereum, a well-known DLT smart contract platform, uses a transaction-based state transition system in a permissionless environment. By 2018, it had been in operation for three years and used PoW as its consensus protocol, with plans to transition to PoS in the future. Ethereum's development is guided by the Ethereum Foundation, developers, and social influencers, and it features the EVM for executing smart contracts (Buterin, 2014; “Ethereum Classic webpage’,” 2021; Wood, 2014).

POA.Network is a permissioned smart contract platform linked to the Ethereum blockchain. It uses PoA as its consensus mechanism and had been in operation for 0.5 years by 2018. The platform is influenced by the POA company and selected validators, and it operates with the EVM (Authority LLC, 2021).

Energy Web Foundation also employs a PoA-based smart contract platform using the Ethereum blockchain, with validators selected by the foundation. Operating on the "Tobalaba" test-net for 0.5 years by 2018, this permissioned system is designed to be influenced by the Energy Web Foundation and its validators, utilising the EVM for smart contracts (Foundation, 2018).

Dfinity is described as a "blockchain supercomputer" designed to host the next generation of software and services. Although it had not yet launched by 2018, it was expected to operate with unknown consensus mechanisms and was influenced by the Dfinity company. Its smart contract capabilities were not fully disclosed (“Dfinity webpage’,” 2021).

Plasma, a proposed framework associated with Ethereum, aims to enable scalable and enforced execution of smart contracts, handling a significant number of state updates per second. Although it had not been implemented by 2018, it was developed by Ethereum's developers and designed to work with the EVM (Poon and Buterin, 2018).

Ethereum Classic, which shares its origin with Ethereum, is another DLT smart contract platform using transaction-based state transitions. Operating as a permissionless system with PoW, it had been in operation for three years by 2018.

The platform is maintained by the ETCDEV team, developers, and social influencers, and it also uses the EVM (“Ethereum Classic webpage’,” 2021).

EOS aims to become a decentralised operating system capable of supporting industrial-scale decentralised applications. Operating in a permissionless environment, EOS uses DPoS as its consensus mechanism. The platform is influenced by the “Block.one” team, developers, and social influencers and features a WebAssembly (Wasm)-based smart contract virtual machine.

Hedera Hashgraph is a distributed computing platform that stores transactions in a graph structure, proposing high transaction throughput capacity. Although it was still in its early stages by 2018, it is designed to operate with PoS consensus. The platform is influenced by Swirlds Inc., Leemon Baird (the patent holder), the Hedera Hashgraph company, and the Hedera Hashgraph Council. The smart contract virtual machine details were not fully disclosed (Baird, 2021; Baird et al., 2021; “Hedera hashgraph webpage’,” 2021).

Hyperledger Fabric, contributed by IBM, is a permissioned blockchain infrastructure managed by the Linux Foundation. It uses BFT/PoA consensus protocols and supports various smart contract virtual machines, including the EVM (“The Linux Foundation, ‘Hyperledger Architecture,” 2021).

Hyperledger Sawtooth, another permissioned DLT system managed by the Linux Foundation, is based on Intel’s "Software Guard Extensions" and employs PoET as its consensus mechanism. The details of its smart contract capabilities were not fully disclosed by 2018 (Rilee, 2021; “The Linux Foundation, ‘Hyperledger Architecture,” 2021).

iOlite is a permissionless platform designed to allow non-programmers to write smart contracts and design blockchain applications using natural language. Although still in its early stages by 2018, iOlite uses PoW as its consensus protocol and is influenced by the iOlite Telegram community, developers, and social influencers. It supports both EVM and Wasm (“iolite webpage’,” 2021).

Lisk is a distributed application platform and blockchain that provides an SDK (Sidechain Development Kit) written in JavaScript. Operating as a permissionless system with DPoS, it had been in operation for two years by 2018. The platform is driven by community developers and social influencers but does not include a smart contract virtual machine (“Lisk webpage’,” 2021).

Neblio is described as a secure, distributed platform built for enterprise applications and services. The platform was still under development as of 2018 and is expected to operate with Proof of Stake consensus. The specific details of its smart contract virtual machine were not disclosed (“Neblio white paper’, 26-Jul-2016,” 2021).

Polkadot is designed to connect private/consortium chains with public/permissionless networks. Operating as a permissionless system, Polkadot was still in its testing phase with the "Krumme Lanke" testnet as of 2018. The platform uses a bespoke consensus protocol and is influenced by the Web3 foundation, with support for both EVM and Wasm (“Polkadot lightpaper version 1’,” 2017; “Polkadot network webpage’,” 2021; Wood, 2016).

Stellar is a smart contract platform where Stellar Smart Contracts (SSC) are composed of transactions connected and executed using various constraints. Operating as a permissionless system for four years by 2018, Stellar uses the "Federated Byzantine Agreement" consensus protocol and is managed by the Stellar Development Foundation. The specific details of its smart contract virtual machine were not disclosed (Foundation, 2021).

In general, applications are found where two or more parties must come to an economic agreement over the status of a thing (or a collection of things) that has a trustworthy digital interface, or where information must be immutably stored (i.e. where it must be provably unchanged with a relatively high degree of certainty). An example might be hashing values (fingerprints) of controller firmware.

2.3 Applications of DLT in Energy Sector

2.3.1 Introduction

DLT holds significant potential in sectors where there are no physical exchanges, such as the financial, banking, and insurance industries. In these sectors, DLT can offer reliable transaction records without requiring physical verification (Luke et al., 2018). As electricity cannot be tracked between two points in an electricity network, electricity markets are typically pooled. This means that electricity sales and purchases are cleared in aggregate on centralised trading platforms, similar to stock exchanges and other financial markets (Luke et al., 2018). As a result, outside of finance, the energy sector is viewed as one of the industries where DLT could have the most transformative and disruptive impact.

Figure 2:13 DLT Maturity evolution

In the energy sector, DLT has the potential to revolutionise how transactions are arranged, recorded, and verified. The traditional centralised model—relying on exchanges, trading platforms, and centralised energy companies—could shift toward decentralised systems where end customers interact directly. The Figure 2:14 illustrates the state of DLT development across various sectors, showing that the financial sector is leading the way, with DLT transitioning from the exploratory stage to the growth stage. The energy sector is closely following, approximately two-thirds of the way through the initial exploratory stage (Frei, 2018).

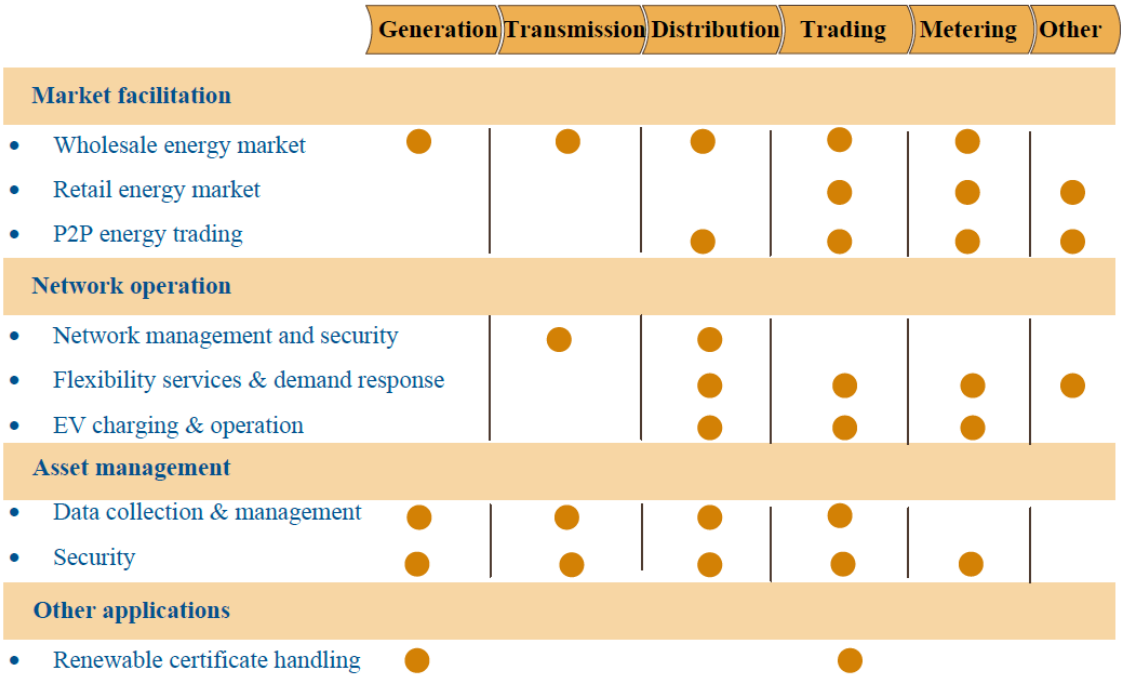


Figure 2:14 DLT use-cases along the energy value chain

2.3.2 Wholesale energy trading

In electricity and gas trading, transactions are typically conducted through an online exchange or via a broker. Traders first consult an index agency to gather pricing intelligence, then each trader independently enters the transaction details into their respective IT systems, known as energy trading and risk management (ETRM) systems. The back offices of both buyers and sellers then retrieve these transaction details from their ETRM systems and exchange data with each other and/or with the broker to confirm and reconcile the trade. This step can be achieved either through an automated confirmation system, such as EFETnet in Europe, or via traditional communication channels like emails, phone calls, faxes, and spreadsheets. The trade is then physically settled through a Transmission System Operator (TSO) and financially settled through a clearinghouse or bank.

This process, which relies on a centralised exchange or broker, can sometimes lead to inefficient communications, resulting in high transaction costs (due to costly exchange and broker fees, and pricing agencies) and operational costs (due to time-consuming reconciliation issues and expensive back-office processes).

DLT offers the potential to significantly reduce transaction costs for trading large volumes by eliminating the need for an exchange or broker. It could also streamline operational processes by directly connecting the trading desks of all parties involved. This would enable DLT-based trading platforms to remove the necessity for brokers and clearinghouses. Additionally, by lowering transaction and operational costs, DLT could make it feasible for participants to trade in smaller volumes (Luke et al., 2018).

An example of DLT's potential in this area is the "Enerchain" pilot project (Burgwinkel, 2016), which aims to reduce costs associated with wholesale energy trading by utilising blockchain technology. Managed by Ponton, a software and energy market automation company, the Enerchain project has developed a proof-of-concept blockchain-based clearing platform for wholesale energy trading that does not rely on a centralised exchange or brokers.

2.3.3 Retail electricity market

DLT has the potential to revolutionise retail electricity markets by automating the "meter-to-cash" process and using cryptocurrency for bill settlement. By enabling the instantaneous settlement of trades, DLT could eliminate the need for traditional energy suppliers and significantly reduce the variable costs associated with payment processing. Some experts even suggest that DLT-based meter-to-cash automation could eliminate the need for wholesale-to-retail intermediaries altogether (Luke et al., 2018).

In addition to cost savings, DLT could enhance the retail customer experience by providing greater transparency into energy supply, charges, and bill components. A DLT-based platform in the retail electricity market could offer customers the ability to enter and exit energy contracts more easily, as well as greater choice in energy supply options.

One example of innovation in this area is a start-up company in Seattle, US, called Drift. Drift is developing a blockchain-based platform that operates as a competitive energy supplier in deregulated markets. The platform leverages DLT, machine learning, and high-frequency trading to connect independent power generators directly with residents and small to medium-sized enterprises. Drift provides bills on

a seven-day cycle, offering detailed information on fees and energy sources. Customers can use a web dashboard to track transactions and choose between zero-carbon energy or the lowest-cost energy options. Notably, Drift operates on a contract-free basis, giving customers flexibility in their energy choices (“Drift Marketplace Inc (US) website’,” 2021).

Another start-up, Grid+, based in Texas, US, is developing an Ethereum-based platform that automates the billing and settlement process. Grid+ aims to provide customers with nearly seamless access to the wholesale energy market by automating these processes. The project employs a two-token model, and an Internet-enabled energy gateway called the Grid+ "Smart Agent." This device, located in the customer's home, primarily functions as an automated payment processing unit, reading from the household's smart meter and paying for electricity usage in real-time (typically in 15-minute to 1-hour intervals, depending on the market). It does so by executing smart contracts on the Ethereum blockchain using "BOLT" tokens, which are securely stored in an e-wallet. A BOLT is a stablecoin representing one dollar's worth of power from Grid (Grid+, 2017).

2.3.4 Peer-to-peer energy trading

Peer-to-peer (P2P) energy trading represents an approach in the demand-side management of power systems, particularly in managing the increasing integration of distributed energy resources (DERs). In P2P energy trading, prosumers—who both produce and consume energy—directly trade energy with each other, aiming for mutually beneficial outcomes. From a power systems perspective, P2P energy trading has the potential to facilitate local energy balance. From the prosumers' viewpoint, it could reduce energy bills (as consumers) and increase income (as producers).

DLT is considered a key enabler for P2P energy trading due to its decentralised, trustworthy, robust, and automated features. DLT could support the development of local marketplaces where energy producers and consumers transact directly on a local scale. By fostering these local markets, P2P energy trading using DLT could alleviate stress on transmission networks, thereby reducing network costs, improving

the economics of small-scale renewables and DERs, and providing customers with greater choice and transparency in energy supply.

In the P2P energy trading system, smart meters play a crucial role as the interface between the electricity system and the blockchain. These meters record electricity generation, imports, and exports, and validate transactions.

Various studies and trials worldwide have explored using DLT to enable P2P energy trading. For example, E. Mengelkamp et al. presented an initial proof-of-concept for a simple local energy market based on DLT (Mengelkamp et al., 2018a). This study involved designing and simulating a local market of 100 residential households using a private blockchain to create a decentralised market platform for P2P energy trading, eliminating the need for a central intermediary. While the study confirmed the effectiveness and economic viability of the DLT-enabled market, it also highlighted the need for further research on the suitability of DLT as the primary ICT for local energy markets, especially regarding its real-life applicability and technological limitations such as resource consumption and scalability.

Another study by J. Kang et al. proposed a localised P2P electricity trading system using a consortium blockchain to enable localised electricity transactions among plug-in hybrid electric vehicles within smart grids (Kang et al., 2017). The study introduced an iterative double auction mechanism designed to maximise social welfare in P2P trading. The consortium blockchain addressed transaction security and privacy concerns, with numerical results based on a real map of Texas validating the proposed methodology.

In a different context, J. J. Sikorski et al. demonstrated a machine-to-machine electricity market within the chemical industry, as part of the fourth industrial revolution (Industry 4.0) (Sikorski et al., 2017). This proof-of-concept implementation employed blockchain to facilitate electricity trading between two producers and one consumer, using realistic data generated by process flow sheet models.

Industry trials have also explored DLT in real-life P2P energy trading scenarios. One of the earliest and most notable projects is the Brooklyn microgrid in New York, USA

(Mengelkamp et al., 2018a). Located on President Street in Brooklyn, this grid-connected microgrid consists of 10 customers—five prosumers with solar panels and five consumers without solar systems. The project established a local P2P energy market using blockchain technology, enabling customers to trade renewable energy directly with their neighbours.

In the Brooklyn microgrid, each customer installs a TransActive Grid element (TAG-e), a hybrid device containing an electricity meter and a computer ("Brooklyn MicroGrid'," 2021). The meter records the net electricity of the customer (whether consumption or surplus) and sends this record to the computer, which acts as an agent in the blockchain. The computers of all the customers form a blockchain-based energy trading platform, where tokenised net electricity is transacted automatically according to pre-written smart contracts that specify trading rules, processes, and conditions. This setup allows customers to trade energy without a centralised intermediary, simplifying the process and reducing energy costs.

Similar trials have been conducted in other countries. In Perth, Australia, a DLT-based P2P solar trading trial began in late 2016, where 10 households bought, sold, or swapped excess solar electricity directly using a blockchain platform that manages multiple trading agreements among prosumers (Vorrath, 2021). In Slovenia, the SunContract platform was launched in April 2018 to support direct trading among customers (Cointelegraph, 2021). Japan is also preparing to launch a similar trial, where a DLT-based platform called Synergy will be deployed on the Plasma network with Ethereum to enable prosumers to trade excess solar power (Richardson, 2021). In the UK, Hackney's Banister House Solar is set to use the blockchain-based P2P energy trading solution Verv 2.0, allowing 40 flats to share lower-cost renewable energy among themselves for free (Bracciali et al., 2018).

2.3.5 Network control and management

With the ongoing decarbonisation of the energy sector, the nature of the electricity grid is undergoing significant changes. The grid is no longer solely dependent on centralised energy generation to meet electricity demand. Instead. Distribution networks are seeing a growing integration of DERs, such as distributed generation (DG) and energy storage. Additionally, smart meters are being increasingly installed

in customers' homes. These changes have made distribution networks more complex due to the inclusion of DERs and digital technologies. Modern Distribution System Operators (DSOs) and TSOs now face the challenges of better understanding the current state of the systems, as well as storing and analyzing vast amounts of data. At the same time, the increased digitalisation of the grid has heightened its vulnerability to cyber-attacks.

DLT has the potential to address many of the challenges in grid management, particularly in distribution networks where decentralised control across multiple timescales, scales, and geographic areas may be necessary. DLT enables the creation of a distributed ledger where data are stored and verified using a linked data structure (i.e., blocks), generated and updated through consensus algorithms, protected by cryptography, and operated via autonomous scripts. By integrating smart contracts, DLT allows transactions to be automatically executed based on predefined rules, such as specifications for quantity, quality, and price.

DLT could significantly improve network management by automatically maintaining verifiable condition data of network assets. Additionally, DLT's inherent redundancy, tamper-proof nature, and lack of a single point of attack make it a robust solution against grid-related cyber threats.

One potential application of DLT in network management involves scenarios where two or more DNOs have different preferences for the control of a device that impacts multiple networks—such as a medium voltage direct current (MVDC) link between two DNO areas. DLT can facilitate a contractual arrangement between these parties, defining the rules for device control. These rules can be encoded in the form of a smart contract. Smart contracts offer the advantage of programming self-enforcing agreements that can automatically dictate the control settings of a shared device.

2.3.6 Flexibility services and demand response

In future power networks, electricity generation will come from multiple sources and flow in multiple directions, providing consumers with greater flexibility and choice. A significant trend is the potential transition from DNOs to DSOs. This transition would involve DNOs taking on responsibilities for managing regional supply and demand

balance through system balancing services and ancillary service markets. The increasing penetration of variable wind and solar generation poses challenges to system operators' abilities to balance short-term supply and demand without resorting to curtailing renewable generation.

DSOs often face situations where there is either too much or too little energy available for balancing demand. This issue is primarily due to the rise in distributed renewable generators, whose intermittent nature complicates energy management. While many distributed energy prosumers can provide upwards or downwards flexibility quickly, the current system underutilises the flexibility of these demands because they take time to respond, or it is too challenging to access small scale flexibility prosumers. The potential benefits of leveraging flexibility services and demand response to balance supply and demand are significant. With DLT, response times could be greatly reduced, as smart contracts can be deployed rapidly, allowing automatic energy agents to adjust energy production by prosumers in real-time. Moreover, smart contracts could enforce penalties on prosumers who fail to meet their commitments.

DLT could play a crucial role in enhancing flexibility services by recording resource availability and automating demand response and DER activities in real-time. A decentralised DLT mechanism would provide transparent, secure, reliable, and timely energy flexibility data of individual prosumers to all stakeholders involved in flexibility services.

An example of DLT’s application in this area is the project initiated by TenneT, a TSO in the Netherlands, in collaboration with Vandebron, Sonnen, and IBM. This project aims to enhance the flexibility services available to the operator by integrating flexible capacity supplied by electric vehicles and household batteries into the grid, particularly for congestion management in the transmission network.

The project utilises IBM Blockchain, which is built on Hyperledger Fabric, a blockchain framework hosted by The Linux Foundation. IBM Blockchain is designed to verify and document the performance of distributed flexible energy devices. This technology is well-suited for connecting multiple parties and a large number of

distributed computing nodes, enabling them to undertake joint actions within a scalable, transparent, and trusted network. The blockchain platform developed by IBM will ensure the verifiability and transparency of transactions involving small-scale batteries and electric cars. It aims to optimise energy distribution across all markets and functions. TenneT will gain insight and the ability to activate flexibility within the energy system, while consumers are empowered to make their flexibility available to the balancing market.

TenneT will test this new concept in two pilot projects in the Netherlands. These projects will evaluate the effectiveness of integrating blockchain technology into the management of distributed flexible energy resources, with the goal of improving the efficiency and reliability of the power grid.

2.3.7 EV charging and operation

The growth of electric vehicles (EVs) has been significant over the past decade, with global annual sales surpassing 1 million for the first time in 2017. As e-mobility continues to rise, it is estimated that by 2030, there will be around 200 million EVs on the roads. To meet the growing demand for recharging these vehicles, a substantial increase in both public and private charging infrastructure has been observed. However, the current EV charging market is highly fragmented, with various apps and cards required to access different charging points, leading to a complex and costly settlement process between EV companies. Additionally, the increasing load on power grids has contributed to a less-than-optimal experience for drivers.

DLT has the potential to improve the coordination of EV charging by simplifying energy payments at charging stations and enabling drivers to make more informed charging decisions based on real-time map and pricing data. An active example of such an initiative is MotionWerk's "Share&Charge" app. In 2016, Innogy partnered with the German blockchain start-up Slock.it to develop a peer-to-peer service that allows EV and charging point owners to autonomously rent out their charging infrastructure without the need for an intermediary. By May 2017, Innogy's Innovation Hub had spun out MotionWerk as a start-up company, with "Share&Charge" as its first product. This app enabled EV owners to charge their vehicles by making digital payments through a mobile application. Charging point

owners (CPOs) could use the app to make their infrastructure available, set tariff structures, and collect fees.

The Share&Charge infrastructure interfaces with a decentralised protocol using the Ethereum blockchain. This protocol integrates CPOs, mobile service providers (MSPs), and the grid, creating a seamless network for EV charging. End-users can easily access charging station maps via their mobile devices.

Until April 2018, the Share&Charge service was available to approximately 1,000 EV owners, with 1,250 private and public charging points registered in Germany. The system utilised an e-wallet and smart contracts on the public Ethereum blockchain as the peer-to-peer transaction layer, including a Euro-based Mobility Token. Share&Charge was the world's first e-mobility transaction platform to leverage blockchain technology. Drawing from end-customer experiences and insights gained from various pilot initiatives in the EU and the US, MotionWerk is currently evolving into an open-source and decentralised digital protocol for EV charging. The aim is to allow charging point operators and e-mobility service providers to fully decentralise their e-mobility assets, thereby simplifying the processes of controlling, payment, and settlement of EV charging.

2.3.8 Data collection and management

In the complex system of an electricity grid, which is governed by physical boundaries and network-specific rules, maintaining a balance between demand and supply is a perpetual challenge. This challenge is further compounded by ongoing industry disruptions such as digitisation, decarbonisation, decentralisation, and electrification.

These disruptions present an opportunity for Distributed Ledger Technology (DLT) to emerge as a foundational technology for managing secured data with integrity. DLT can securely and transparently integrate data from the grid's edge to the cloud, allowing for data to be collected, stored, and later examined for reconciliation purposes. A DLT-based platform, with its tamper-proof audit trail of data, can be utilised by a central organisation or multiple third parties, each sharing different parts of the platform with various stakeholders. This approach enhances transparency in a

secure manner and often eliminates the need for intermediaries, whose primary role is to create trust in a chain of transactions or information exchanges. The single-source-of-truth ledger that DLT provides enables significant efficiency and productivity gains by eliminating the need for reconciliation. It also facilitates transactions in a more efficient, economical, and intelligent manner.

UK start-up Electron, founded in November 2015, is dedicated to developing blockchain-based platforms for the energy industry. Electron has created an ecosystem of blockchain platforms, including systems for asset registration, flexible trading, and smart meter data privacy. One of the company's most notable achievements is its work on customer utility switching. Using dummy data to test more than 55 million supply points, Electron demonstrated that customers could be switched from one supplier to another in as little as 15 seconds ("Electron'," 2021). Electron's work exemplifies how DLT can streamline processes within the energy sector, paving the way for faster, more reliable, and more secure transactions and data management.

DLT offers significant potential to enhance security in the energy sector, thanks to its distinctive method of recording and processing data. As global demand for energy continues to rise, power companies face increasing pressure to ensure their operations are both secure and efficient. DLT can contribute to these goals by streamlining the process of transporting power, ultimately saving time and money. Moreover, DLT has the potential to fundamentally alter the way electricity is produced and consumed.

One of the key benefits of DLT is its ability to facilitate secure, practical transactions between buyers and sellers through the use of smart contracts. These contracts eliminate the need for third parties by automating transaction processes, signaling to the system when to initiate specific transactions. This automation not only increases efficiency but also reduces the risk of human error and potential fraud. Decentralised storage of transaction data is another major advantage of DLT, as it enhances security by reducing reliance on central authorities. This decentralised approach ensures that transaction records are more resilient to tampering and cyber-attacks, creating a more robust and secure energy infrastructure.

Additionally, DLT can provide a more efficient system for evaluating energy sources and their impact on consumer prices. By offering a transparent and secure platform for data analysis, DLT enables power companies to make more informed decisions about energy production and distribution. This, in turn, leads to better services and outcomes for consumers, as improved technology translates into more reliable, cost-effective energy solutions.

2.3.9 Renewable certificate handling

Recording renewable certificates presents several challenges, including the costly reliance on manual audit practices, limited geographic scale, and centralised, opaque management. These challenges can lead to high transaction costs and even instances of fraud. One potential solution is the tokenisation of renewable attributes (certificates) and their storage using DLT. By storing the creation and transactions of environmental attributes on blockchains, the need for a central verification agency can be eliminated, provided there is an appropriate governance system in place. Data stored on a blockchain can be made accurate and secure, reducing the risks associated with traditional systems.

An example of a project that addresses these challenges is SolarCoin, a solar-incentivising cryptocurrency. SolarCoin aims to reduce audit costs, improve transparency, and enhance liquidity for solar-derived credits. SolarCoins are awarded to solar generators after they submit claims of generation to the SolarCoin Foundation or an affiliate organisation. These claims can also be generated automatically by smart meters, and all transactions are recorded and visible on the SolarCoin blockchain. As of March 2018, SolarCoins had been distributed in 58 countries, with growing demand for the cryptocurrency intended to incentivise further renewable generation.

Another innovative initiative is IDEO CoLab's integration with Nasdaq’s Linq platform and IoT (Internet of Things) company Filament’s hardware, which uses digital sensors with blockchain capabilities to issue renewable energy credits (RECs) to producers for each kilowatt-hour (kWh) their solar panels generate. This pilot project aims to enable small solar producers to easily track, verify, and trade their generated

power. By leveraging blockchain technology, these efforts seek to simplify and secure the process of managing renewable energy credits, ultimately supporting the growth and sustainability of renewable energy markets.

2.4 Information handling

2.4.1 Information flows

The transition to Smart Grids, along with the decentralisation of responsibility for characteristics such as system stability, is likely to introduce greater complexity into the electricity system. Currently, in the Great Britain (GB) system, the procurement of balancing services can be visualised as a network of contracts with a single hub—the National Electricity System Operator (NESO)—and spokes that connect to the various parties providing these services, as illustrated in Figure 2:15. This centralised structure simplifies the management of contracts and responsibilities.

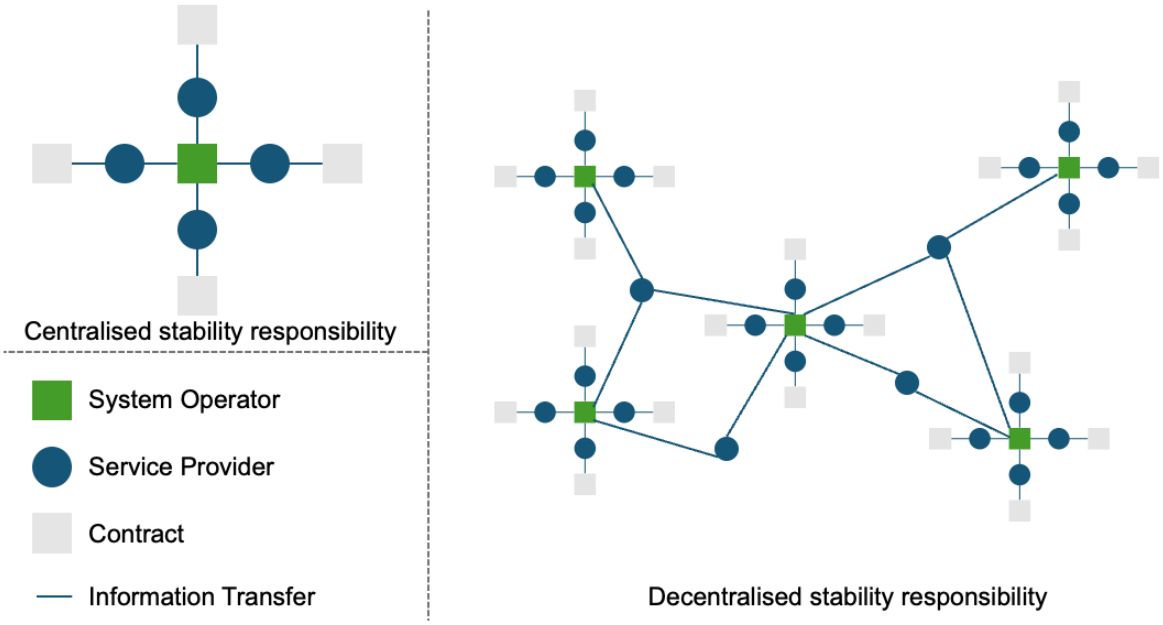


Figure 2:15 Increased complexity of the contract network for balancing commitments under a DSO based system

As the contractual relationships within the electricity distribution system evolve, the role of the DNO is also undergoing a transformation. In future systems where multiple parties share responsibility for network stability, the contractual landscape becomes significantly more complex. The contract graph will no longer consist of a simple hub-and-spoke model but will instead feature numerous interconnections between different parties, each with their own responsibilities and agreements. This

increased complexity poses a challenge to the industry in terms of establishing codes of practice and standards that ensure the system remains resilient. As the network of contracts expands, it will be crucial to develop robust frameworks that facilitate coordination and maintain system stability across the more intricate web of relationships and responsibilities. This evolving landscape not only necessitates new standards and practices but also prompts a fundamental shift in how distribution networks are managed and operated.

The transition of the DNO role to that of a DSO has garnered significant attention in recent years. This shift is viewed as a possible response to various societal and technological changes, including the need to decarbonise the electricity sector, a growing desire for greater community involvement in electricity supply, the rise of electric heating and transport, the increasing deployment of smart meters, and the expanding potential of power electronics and automation. Another relevant technological trend is the decentralisation of information control through DLT, which could reduce the need for intermediaries in the electricity system.

To explore potential applications of DLT in a DSO-based system, this literature review examines how information is created and transferred within the current and possible future electricity systems in GB.

The literature on DSOs highlights several recurring themes:

- Regulatory Change of Responsibilities: One key theme is the idea that DSOs could take on some responsibility for system stability, a role traditionally managed by centralised entities.
- Market Facilitation: DSOs are envisioned as neutral market facilitators that provide localised system balancing. This could involve facilitating various markets, such as those for energy, ancillary services, and flexible connections.
- Digitisation: The transition to DSOs involves increased use of digital monitoring and control to achieve higher levels of flexibility through enhanced visibility and active network management. In some cases, this might also include increasing public transparency regarding the state of the network.

- Increased Interaction with Neighbouring System Operators: As DSOs assume more responsibility, they will need to form more complex agreements with neighboring system operators to ensure overall system stability.
- Increased Interaction with Demand and Distributed Generation: DSOs will also need to establish more sophisticated agreements with demand and generation sources connected to the distribution network. This interaction is crucial to maintaining system stability as the network becomes more distributed and complex.

These themes collectively illustrate the evolving role of DSOs as they navigate the challenges and opportunities presented by the ongoing transformation of the energy sector.

DLT-based smart contracts have been identified as having potential applications in scenarios where two or more parties need to reach an agreement based on digital events and the status of digital assets. However, it is important to note that DLT-based smart contracts are currently not well-suited for real-time operations due to the time required by the DLT consensus mechanism. Therefore, potential applications of DLT-based smart contracts should involve distinct procedures that occur before and after a commitment is delivered.

The processes that occur before a commitment is fulfilled can be conceptualised as "negotiation"—the phase where parties come to an agreement based on their expressed preferences. The processes that occur after a commitment has been made, which involve verifying that the agreed commitment has been delivered and facilitating the transfer of value, can be categorised as "settlement." As such, the points in a process where negotiation or settlement occurs are prime candidates for the application of smart contracts.

To effectively identify where DLT might be usefully applied in the power system, a good starting point is to map the contracts between participants in the system and evaluate the extent to which these contracts can be digitised in a trustworthy manner. This involves defining the participants, identifying information sources, and

understanding the information exchanges that occur. With these definitions in place, the next step is to map the contractual links between parties in both the current and prospective DSO-based power systems in GB. This approach helps in identifying specific areas where DLT-based smart contracts could enhance efficiency, security, and transparency in the power system.

Information exchange interfaces are essential mechanisms that facilitate the transfer of information between parties in various systems, including the energy sector. These interfaces exist at the boundary between different entities and may involve the application of conditional logic to the exchanged information. For example, a rule such as "if X, then pay Y" might be implemented to manage transactions or agreements. Since this logic must be trusted by all parties involved, these interfaces present significant potential for the application of smart contracts. Identifying these information exchange interfaces is a logical starting point for exploring the potential uses of DLT in such systems.

Information exchanges can be broadly categorised into two main types: contractual exchanges and data exchanges. Contractual exchanges define the mechanisms for transferring information between decision-makers to form and execute agreements. The ENA (Energy Networks Association) Open Networks Project is a collaborative initiative aimed at transforming the way electricity networks operate in the UK, by enabling a more flexible, low-carbon energy system. It focuses on the development of new frameworks, standards, and innovations to facilitate the transition from DNOs to DSOs, enhancing the coordination and integration of DERs.

This category aligns with the "contractual" classification used in the ENA Open Networks project and includes three specific subtypes. The first is negotiation, which refers to the process by which decision-makers reach an agreement and commit to specific future actions or behaviours. The second subtype is settlement, which involves the implementation of rules that govern the transfer of value between parties based on a prior agreement. The third subtype is governance, which concerns the transfer of information from regulatory bodies to other participants, constraining or guiding their decisions through standards and guidelines.

The second main type of exchange, known as data transfer, exchanges, defines the mechanisms for transferring information to fulfil contractual or regulatory obligations. This category is also akin to the "information" classification from the ENA Open Networks project and includes two subtypes. The first is operational exchanges, which involve the transfer of information between parties for the purpose of network operation, ensuring compliance with contractual or regulatory obligations. The second subtype is planning exchanges, which pertain to the transfer of information for network planning, also to meet contractual or regulatory obligations.

To effectively identify where DLT could be applied, it is crucial to map these information exchange interfaces within the current and potential future electricity systems. By examining the flow of information between participants in these systems, it is possible to pinpoint specific areas where DLT-based smart contracts could automate and enhance the efficiency of these exchanges.

For instance, in a market system, an information exchange interface diagram might illustrate the flow of information from participants to information exchanges. The processes occurring within these information exchanges are those that could potentially be implemented as DLT-based smart contracts. Conceptually, these smart contracts could function as autonomous intermediaries, managing the negotiation, settlement, and governance processes between participants. By automating these processes, DLT-based smart contracts could improve the efficiency, transparency, and security of information exchanges in complex systems such as the electricity grid.

2.4.2 Security and DLT vulnerabilities

As power systems gradually change into cyber-physical systems (systems with integrated computing, sensor and control hardware) the field of cyber security becomes increasingly critical. There is an active research community around the subject of cyber security in power systems and, more generally in cyber physical systems (He and Yan, 2016). This literature review presents security and hardware aspects when DLT is applied in power systems.

The integration of DLT in power systems presents an opportunity to address various security vulnerabilities inherent in cyber-physical systems. Vulnerabilities across six layers have been identified: governance, economic rules, data storage and communication, software, hardware interface (sensors and controllers), and the physical power system. A comprehensive literature review was conducted to identify example vulnerabilities and potential mitigation techniques using DLT.

- Governance Layer: One significant vulnerability in this layer is regulatory capture, where regulatory bodies may be unduly influenced by industry players, potentially compromising system operations and increasing costs. To mitigate this, DLT can be structured to ensure that regulatory capture does not impact system operations or introduce additional costs, as suggested in literature (Constantinides et al., 2018).
- Economic Rules Layer: Errors in the implementation and storage of economic rules pose another vulnerability. DLT smart contract platforms offer a solution by automating the enforcement of these rules, ensuring accuracy and reliability.
- Data Storage and Communication Layer: The manipulation or corruption of stored or communicated data is a critical vulnerability in power systems. DLT can address this through data protection frameworks that enhance the integrity and security of data, making it more resistant to tampering (G. Liang et al., 2018).
- Software Interface Layer: Software vulnerabilities, whether caused by malicious intent or accidental corruption, can compromise system operations. Independent vetting of software and smart contracts can help mitigate these risks, ensuring that the software functions as intended (Destefanis et al., 2018; Jia, 2018).
- Hardware Interface Layer: The corruption of data collection processes or hardware instructions at the metering and controller level represents a significant vulnerability. Verification of firmware, as part of a DLT framework, can enhance the security and reliability of these interfaces (Banerjee et al., 2018; X. Liang et al., 2018).
- Physical Power System Layer: Complex control interactions emerging from economic rules can destabilise the power system. DLT can be used to design economic rules and system arrangements that minimise or neutralise these complex interactions, thereby enhancing system stability (Thomas et al., 2019).

Among these vulnerabilities, those related to data storage and communication are particularly critical. Modern power systems increasingly rely on advanced communication and control technologies, which impose significant demands on the robustness, efficiency, and security of the underlying information infrastructure. The deep integration of cyber and physical resources means that cyber-attacks can have severe consequences, including misleading decision-making in control centres, financial loss, or even blackouts. Common types of cyber-attacks include false data injection, denial of service (DoS), data framing attacks, and cyber topology attacks (Kim and Tong, 2013; Liang et al., 2019).

While various methods have been developed to detect and defend against cyber-attacks in centralised data communication and storage systems (He et al., 2017; Liu et al., 2017) the wide geographical distribution of meters and sensors in modern power systems necessitates the use of distributed security technologies. DLT, with its intrinsic features of decentralisation, verifiability, integrity, and redundancy, offers a promising approach to enhancing the self-defensive capabilities of power systems. A blockchain-based data protection framework proposed by (G. Liang et al., 2018) for modern power systems exemplifies this approach. Their framework involves reconfiguring the Supervisory Control and Data Acquisition (SCADA) system to include a distributed information gathering and storage mechanism. This system would consist of geographically distributed meter-node networks connected through a private blockchain, ensuring that all collected data are stored in a tamper-proof ledger.

Despite the potential of DLT to enhance security, it is not without its own vulnerabilities. Consensus mechanisms in DLT, for instance, can be compromised by a 51% attack, where a single party or coalition gains control over the majority of the network's mining power, allowing them to introduce false transactions. Other vulnerabilities include double-spend attacks, eclipse attacks, liveness attacks, and balance attacks, all of which can undermine the integrity of the blockchain (Li et al., 2017):

- DoS Attacks: DLT systems are also susceptible to DDoS attacks, where servers are overwhelmed with connection requests, reducing their ability to respond to legitimate users. While DDoS attacks are more challenging to execute against

DLT, they remain a viable threat, as demonstrated by a 2016 incident where the Bitcoin network was nearly brought to a halt by a flood of spam transactions (He et al., 2018).

- Smart Contract Vulnerabilities: Smart contracts, essential to DLT operations, are vulnerable to both accidental and deliberate coding errors, which can result in financial loss or loss of control over processes. Mitigation strategies include implementing methods for self-replacement or destruction of flawed contracts, conducting thorough audits, and potentially using insurance to cover losses (Vessenes, 2022).
- Key Generation Process Vulnerability: The security of DLT interactions heavily depends on the secure generation and handling of cryptographic keys. Vulnerabilities in the key generation process, such as flaws in random number generation or the integrity of the computer generating the key, can compromise the entire system. Best practices include using well-vetted software on isolated computers and employing "air gaps" to separate key storage from network-connected devices.
- Handling of Private Keys: Private keys, akin to passwords, must be securely managed to prevent unauthorised access. Different methods of handling private keys, such as hardware wallets, paper wallets, brain wallets, multi-signatory accounts, and reliance on trusted third parties, offer varying levels of security and convenience. Each method has its own vulnerabilities, and organisations must choose their approach based on risk tolerance and technical capability.
- Social Engineering Vulnerabilities: Finally, social engineering poses a significant risk when individuals or groups interact with smart contracts. Attackers may attempt to manipulate decision-makers to influence the outcome of transactions. Addressing social engineering vulnerabilities requires a combination of robust business processes and ongoing education to enhance awareness and resistance to such attacks (Byres et al., 2014).

2.4.3 Hardware

In general, there are no specific computing systems required for interacting with DLT systems. Interaction with a DLT system could involve activities such as monitoring the system for a particular transaction so that the computer can trigger a specific

action, for example, "X has paid Y, so do Z." Another common interaction is the broadcasting of a signed transaction to the DLT network to update the shared data object. Conventional computing approaches are sufficient for these interactions, and much of the software developed by the DLT industry is designed to operate across standard consumer devices. Currently, there is significant activity in the development of software for interacting with public blockchains, and the same software can generally be applied to permissioned DLT systems as well.

There are no significant special hardware requirements for validating the state of a DLT system. Typically, this validation process involves downloading a full copy of the blockchain and transaction history and then verifying the validity of declared blocks. This task can be performed on a standard personal computer. To enhance reliability and avoid a single point of failure, organisations may choose to duplicate validation nodes across different locations, mitigating the risk of interruptions due to local communication network faults.

Organisations or individuals wishing to participate in the consensus mechanism of a PoW system may opt to use Application Specific Integrated Circuits (ASICs). These are specialised systems optimised for a particular hash algorithm, such as SHA-256 used by Bitcoin, making them highly efficient for mining specific cryptocurrencies. ASICs are favored because they offer a significantly lower energy cost per hash compared to alternatives like Graphics Processing Units (GPUs) or standard PC Central Processing Units (CPUs).

Due to their specialised nature, mining has become an activity primarily concentrated in large mining farms, which are facilities that house numerous ASICs dedicated to cryptocurrency mining. This centralisation has led to a situation where a relatively small number of coalitions control the majority of the mining power, as illustrated in various industry reports and analyses. The concentration of mining power within a few parties in public DLT systems, such as Bitcoin, raises concerns about the potential for a 51% attack, where a single entity or coalition controls more than half of the network's mining power and can potentially manipulate the blockchain.

To mitigate the risk of a 51% attack, parties relying on public DLT systems might consider using specialised mining hardware as a form of insurance. This hardware could be activated in response to signs of a potential attack, thereby contributing additional mining power to the network to protect its integrity. However, it is important to note that even in this scenario, the energy costs associated with manufacturing and shipping the hardware are still incurred, regardless of how frequently the hardware is used.

A review of energy-related DLT demonstration projects has revealed a growing trend toward integrating DLT system monitoring computers, private key storage, and digital signature hardware with metering and control systems. Typically, this integration involves a computer or a small device with control and data processing capabilities connected between end users and the DLT network. Two notable demonstration projects exemplifying this trend are the Brooklyn Microgrid project in New York, USA, and the Grid+ project in Texas, USA.

The Smart Agent prototype, used in the Grid+ demonstration project, interconnects with the customer's smart meter and home energy management system via wireless communications, such as home Wi-Fi. It passes metering and control data into the DLT network through the internet and manages the customer's energy account, including billing and balance monitoring. The Grid+ project has developed multiple generations of the Smart Agent prototype, each iteration refining its capabilities and integration with the DLT network.

In both the Brooklyn Microgrid and Grid+ projects, a computer or small device with control and data processing functions is typically connected between end users and the DLT network. This device, sometimes integrated with a smart meter, facilitates the flow of information from end users to the DLT network, potentially acting as passive nodes within the network. This integration underscores the practical application of DLT in modern energy systems, enhancing the efficiency, transparency, and security of energy transactions and management.

2.5 Summary

Initially, the literature review introduces DLT and its potential to revolutionise the energy sector, particularly in the context of flexibility markets. Flexibility markets are crucial for managing the supply and demand of electricity, especially with the increasing integration of renewable energy sources. The review underscores the importance of digitalisation in these markets, which is vital for the UK's decarbonisation efforts. DLT, with its decentralised and tamper-resistant nature, offers significant advantages in creating transparent, efficient, and secure energy markets. The review critically examines existing research, pilot projects, and real-world implementations to assess how DLT can support the digitalisation of these markets. It also identifies the current challenges and gaps in knowledge that need to be addressed to fully harness the potential of DLT in the energy sector.

A key focus of the review is on the technical features of DLT that make it suitable for various applications, including public verifiability, transparency, privacy, integrity, and redundancy. These features allow DLT systems to operate without a central authority, ensuring that data is securely shared and verified across a distributed network of nodes. The review contrasts DLT with traditional centralised databases, highlighting how DLT’s distributed nature enhances data security and trustworthiness. The discussion also covers the distinctions between permissionless and permissioned DLT systems. Permissionless systems, like Bitcoin, are open to all participants, while permissioned systems restrict access to authorised entities. This differentiation is crucial in understanding how DLT can be applied in different contexts within the energy sector, from public energy trading platforms to private, secure energy management systems.

The evolution of DLT is traced from its origins with the introduction of Bitcoin in 2009, marking the beginning of DLT 1.0, which focused primarily on cryptocurrency transactions. The review then transitions to DLT 2.0, characterised by the introduction of smart contracts and DApps. These innovations expanded the use of DLT beyond simple transactions to more complex applications, including the automated execution of contracts and the management of digital assets. DLT 3.0, the current stage, is focused on extending these applications into new areas, such

as the energy sector. The review highlights how these advancements are being applied to create more dynamic and resilient energy systems, particularly through the use of smart contracts to automate and secure energy transactions.

The review concludes by exploring the practical applications and implications of DLT in the energy sector. It discusses how DLT can streamline various processes, such as peer-to-peer energy trading, grid management, and demand response, by enabling more efficient and secure transactions. The use of DLT to manage energy data and automate interactions between energy producers, consumers, and distributors is seen as a key innovation that could reduce costs, increase transparency, and improve the overall efficiency of energy markets. However, the review also cautions against the challenges and risks associated with DLT, such as scalability issues, the need for robust consensus mechanisms, and the potential for regulatory hurdles. It emphasises the importance of ongoing research and pilot projects to better understand these challenges and develop solutions that can fully realise the benefits of DLT in the energy sector.

With the considerations that the current available literature of DLT applications in the energy sector is exhaustive, the following gaps have been identified which build on the motivation of the technical work conducted in the thesis.
Firstly, even if there are numerous publications which showcase DLT-based frameworks or mechanisms for flexibility markets, there is a lack in the literature of a framework that proposes a viable framework for the flexibility markets in GB. As GB is leading on the development of flexibility services, with innovative business models across both local and national flexibility markets, this presents a fundamental opportunity to explore a viable DLT-based flexibility framework for the GB case. Additionally, the work explored in the literature generally tests the viability of the framework using Ethereum, as it is the most comprehensive DLT for such developments to date. Recently, Hedera hashgraph has seen quick advancing in the functionality of the ledger, thus development is possible there. Therefore, there is a need for the framework to be tested on multiple ledgers, and for the results to be compared to further understand ledger viability.

Secondly, with the up speed in deployment of small-scale flexibility prosumers, generally accessed through aggregators or suppliers, there is a need of fundamentally being able to settle flexibility transactions without putting a prosumer at large cyber security risks. The literature suggests various models for dealing with the matter, but none of the source's investigating are proposing a ZKP-based flexibility settlement mechanism which is viable for GB.

Lastly, there is a high concentration of work in the literature expressing the need of a just, inclusive and customer-centric Net Zero journey, and certain sources do explore potential financial mechanisms to upscale the adoption of flexibility assets, such as EVs, PVs (Photo Voltaic) or heat pumps. But there is a lack of exploration over how blockchain and ROSCAs can be used together to build a more cost-effective financial product. Thus, it will be valuable for the community for this use-case to be tested, again for the UK.

3 A distributed ledger technology-based framework for flexibility services market facilitation in electricity distribution networks

This chapter presents a decentralised framework for managing flexibility markets in electricity distribution networks. The framework leverages DLT to automate processes such as bidding, contracting, and settlement through smart contracts. A private, permissioned DLT network has been designed for controlled participation, ensuring cybersecurity and access efficiency. The smart-contract architecture has been designed to encode the market rules and to automate transactions, thus reducing the need for manual interventions or intermediaries. To test and proof the logic correctness of the framework, a simple market clearing algorithm has been developed which checks the stability of voltage magnitude at each busbar, expressed in p.u. To evaluate the efficiency of the framework against different DLT architectures, a set of performance metrics has been defined. Thus, the work evaluates latency, transaction fees, and scalability using Ethereum and Hedera Hashgraph.

3.1 Introduction

The industrial revolution, driven by the widespread use of fossil fuels, has had a profound impact on the world's climate. The burning of these fuels has led to a significant increase in greenhouse gas emissions, resulting in global warming and climate change. In order to mitigate the effects of climate change and reach Net Zero emissions, it is essential to transition to low-carbon electricity supply paradigm (*Intergovernmental Panel on Climate Change (IPCC*, 2014).

In the UK, the government has set ambitious targets for reducing greenhouse gas emissions in order to combat climate change. The GB Climate Change Act of 2008 established a legally binding target to reduce carbon dioxide (CO2) emissions by at least 80% by 2050, compared to 1990 levels (Parliament, 2008). More recently, the GB government has set an interim target of reducing CO2 emissions by 78% by 2035, and to reach Net Zero by 2050 (*Department for Business, Energy & Industrial Strategy (BEIS*, 2020). This will not be an easy task, but it is essential for the GB to meet these targets in order to play its part in tackling climate change and ensuring a sustainable energy future. Additionally, the GB government has committed to decarbonise the electricity grid by 2030 (*Department for Business, Energy & Industrial Strategy (BEIS*, 2020).

3.1.1 A smart, flexible energy system.

As both supply and demand become significantly more dependent on weather and increasingly volatile, there will be times when the electricity system has an excess of renewable generation and other times when it has too little (ESO, 2021). To manage this, GB must have access to dispatchable power to accommodate these fluctuations in supply and demand, which is the essence of flexibility. By enabling greater flexibility across the entire electricity system, it will be possible to minimise the cost to consumers of decarbonising the UK’s electricity network and ensure a reliable power supply.

In this context, a smart and flexible electricity system needs to be developed, which enables real-time monitoring and control functionalities and integrates various DERs such as small-scale solar and wind power, distributed energy storage, as well as

EVs, heat pumps and other flexible demand. This allows for the efficient use of available resources, the ability to respond quickly to changes in demand, and the management of the intermittent nature of variable renewable energy sources to ensure a stable and reliable power supply (Catapult, 2019).

3.1.2 The role of flexibility in a Net Zero system

Flexibility plays a crucial role in the smart and flexible energy system for achieving Net Zero transition (*International Energy Agency (IEA*, 2021). Flexibility can be defined as the ability of the system (or part of the system, such as generation, storage and demand) to adjust its electricity supply or demand level according to the need of the system or external signals (such as pricing signals). Without flexibility, a higher amount of over-generation and load curtailment is expected, which leads to higher operational and balancing costs (London and Trust, 2016). Furthermore, on one hand, in a low flexibility scenario, the need for additional power plants, and other forms of back-up generation, can significantly increase the investment costs (Consulting, 2017). On the other hand, a high flexibility scenario allows for the integration of more renewable energy, reducing the need for fossil fuel and back-up generation, and ultimately reducing the costs. Therefore, flexibility is a key factor in achieving a cost-effective sustainable energy system, and is essential for meeting the ambitious climate targets of the GB and the world (Climate Change, 2019). Figure 3:1 illustrates the difference in system costs in a high and low flexibility scenario.

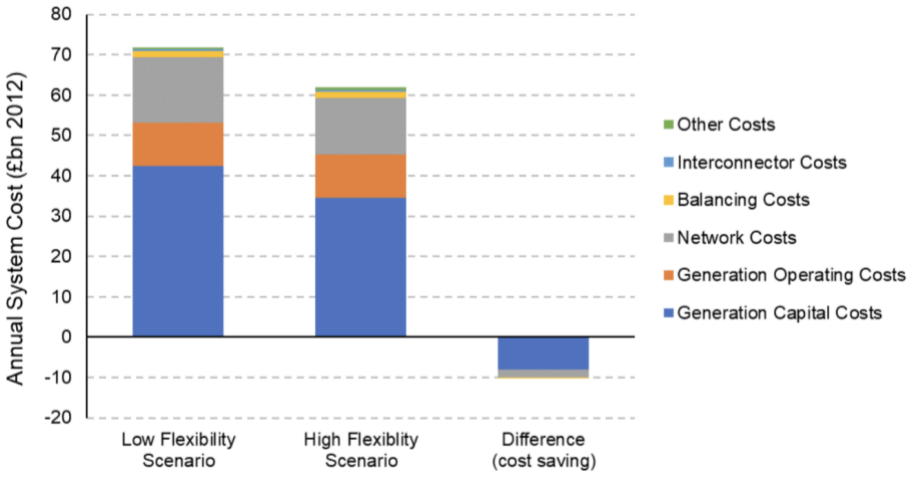


Figure 3:1 Illustrative system costs in 2050 (ESO, 2021)

The GB Government has been actively working to increase the flexibility of the electricity system. In recent years, the government has implemented multiple measures, including the ESO's demand flexibility service, the roll-out of smart meters, and the creation of a capacity market to ensure that there is sufficient generation capacity to meet demand (ESO, 2021). The government has also invested in energy storage projects and EV infrastructure.

In the future, the government plans to continue to focus on increasing electricity system flexibility through the development of new technologies, such as advanced energy storage systems, and through the promotion of demand-side management, to enable customers to adjust their energy usage in response to changes in supply and demand (*Department for Business, Energy & Industrial Strategy (BEIS*, 2020). They are also planning to invest in interconnection projects to increase the amount of electricity that can be imported and exported across borders, which will help to balance supply and demand and improve the security of the energy supply.

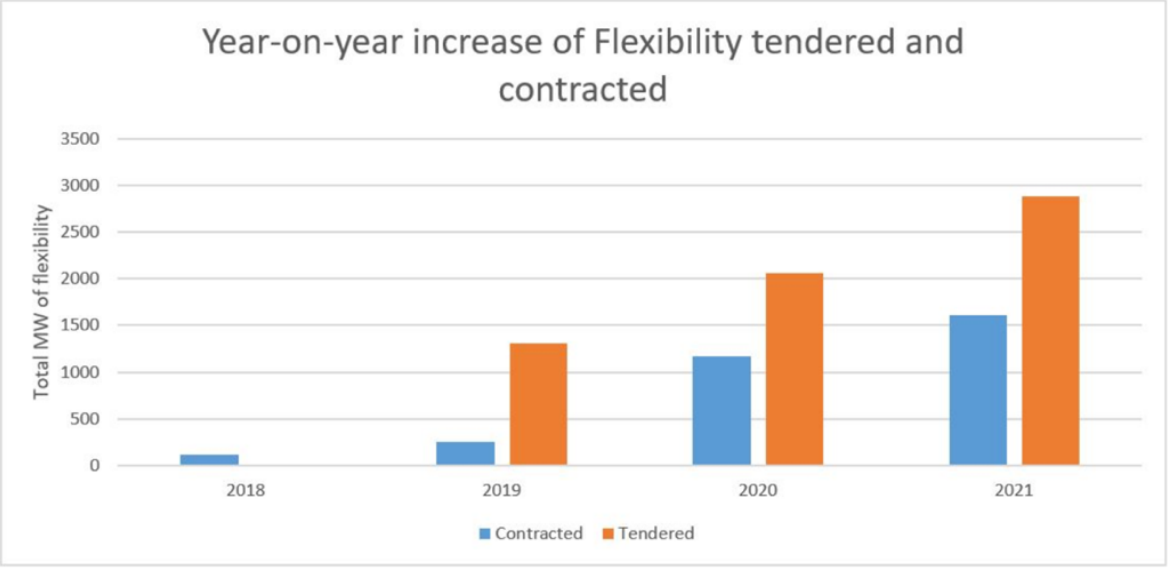


Figure 3:2 Year-on-year increase of flexibility tendered and Contracted (ENA, 2021)

3.1.3 Overview of ENA's Open Network Project on flexibility services.

In the UK, the ENA is a trade association representing the companies that operate the gas and electricity networks in GB and Northern Ireland ("Energy Networks Association (ENA," 2022). Among the ENA's members there are DNOs, and they are responsible for the distribution of electricity to homes and businesses. One of the key

initiatives of the ENA is the Open Networks project, which aims to create a more flexible and efficient energy system by leveraging new technologies and innovative business models (“Energy Networks Association (ENA,” 2021).

As part of this initiative, DNOs have made a commitment to developing flexibility services, which are designed to help balance supply and demand in the electricity system (“Energy Networks Association (ENA,” 2018). These services include the ability to store and shift energy, as well as demand-side management, which enables customers to adjust their electricity usage in response to changes in supply and demand. This can include time-of-use tariffs, which encourage customers to shift their electricity usage to times of the day when there is more renewable energy available, and demand response programs, which pay customers to reduce their electricity usage during periods of high demand. Through these flexibility services, DNOs are working to ensure that the electricity system is able to adapt to the increasing penetration of renewable energy and the changing energy needs of customers.

The ENA has developed a set of 6 steps for delivering flexibility services (Association, 2019), which are designed to help DNOs and other industry stakeholders to implement flexibility services in a consistent and effective manner. These steps are:

- Assessing the need for flexibility: This step involves identifying the key drivers of flexibility, such as the increasing penetration of renewable energy, and assessing the potential benefits and costs of different flexibility options.
- Defining the flexibility services: This step involves specifying the technical requirements and performance characteristics of the flexibility services that will be provided.
- Designing the flexibility services: This step involves designing the flexibility services and the necessary market and regulatory frameworks to support their delivery.
- Implementing the flexibility services: This step involves the actual deployment of the flexibility services, including the necessary infrastructure and systems.

- Operating the flexibility services: This step involves the ongoing management and operation of the flexibility services, including the monitoring and control of the electricity system.
- Evaluating the flexibility services: This step involves assessing the performance of the flexibility services and identifying opportunities for improvement.

The DNOs are expected to engage with their customers, including local authorities, energy service companies and technology providers, to identify and develop solutions that can provide flexibility services to the electricity system. Additionally, DNOs are also committed to providing transparent and fair access to their networks, to support the integration of new technologies and business models. The DNOs also need to ensure that they have the right regulations and market frameworks in place to support the integration of flexibility services, and they are required to report on their progress towards meeting these commitments to Ofgem, the GB energy regulator.

There are several challenges that need to be addressed in order to successfully build and develop flexibility services in electricity distribution networks. Some of the main challenges include:

- Net Zero Focused: The development of flexibility services must be undertaken in a manner that helps the electricity grid manage the intermittency of distributed generation, weather dependency, and decentralisation.
- Security: The integration of flexibility services must not compromise the security of the electricity system. This includes ensuring that the system is able to withstand disruptions and that there are adequate measures in place to prevent and respond to cyber threats.
- Redundancy: The integration of flexibility services must be done in a way that ensures that the electricity system is able to continue operating even in the event of a failure of a key component of the system. This includes ensuring that there are adequate backup systems and procedures in place.
- Transparency: The contracting of flexibility services must be done in a transparent and fair manner, with clear and consistent rules for access to the

grid and for the provision of flexibility services. This includes ensuring that there is open access to data, and that there is a level playing field for all participants.

These challenges require to be addressed through collaboration between different stakeholders, including the government, regulators, utilities, and private sector entities, and through the development of new technologies and business models.

It also requires a regulatory framework that facilitates the integration of flexibility services, provides for fair and transparent access to the grid, and ensures that the costs and benefits of flexibility services are allocated in a fair and efficient manner.

3.1.4 DLT as a potential candidate to support the flexibility market

DLT is a decentralised digital system that uses a network of computers to record and validate transactions in a secure and transparent manner (Andoni et al., 2019). This makes it an overall good candidate to support the digital backbone of flexibility market as it can provide a secure, transparent and tamper-proof digital environment. In this context, flexibility markets are defined as platforms or systems where various participants can trade and procure flexibility services.

DLT can provide several benefits and address the four challenges in contracting for flexibility services (Kang et al., 2017), including:

- Environmental: DLT can be used to track and verify the source of energy used for flexibility services, providing a transparent and auditable record of the environmental performance of flexibility services.
- Security: DLT can provide a secure platform for the recording and validation of transactions and can be used to implement smart contracts that automatically execute transactions based on pre-defined rules.
- Redundancy: DLT can be used to provide a decentralised and distributed system that can continue to operate even if a single node fails.

- Transparency: DLT can provide a transparent and tamper-proof platform for the recording and validation of transactions, which can be used to ensure fair and transparent access to the grid and flexibility services.

Therefore, DLT can enable the automation of processes, improve the reliability and transparency of data, and increase the trust among the parties involved in the flexibility market.

DLT can help shift flexibility markets from an independent and centralised system towards an interconnected and decentralised model (Sousa et al., 2019a). This is possible as DLT allows for the creation of a decentralised network of participants that can securely and transparently share information and transact with one another. This is beneficial in overcoming the limitations of traditional centralised systems, such as a lack of transparency and a lack of trust among participants.

The native architecture of DLT can then allow participants to directly transact with one another without the need for a central intermediary. Even in single buyer markets, this enhances competition and reduce costs in the market. Additionally, through the use of smart contracts, DLT can enable the automatic execution of transactions based on predefined rules, which can help to improve the efficiency and reliability of the market.

Furthermore, depending on the development of regionality of markets in further energy regulations, DLT can allow different participants to share and trade flexibility across different regions, balancing supply and demand in a more efficient way. this further supports the need of change in expensive centralised infrastructure and increase the resilience of the system against disruptions. This is of course, dependent on how regulations will evolve in the future.

3.2 The current flexibility market in the GB

3.2.1 Flexibility services developed by ENA

ENA in the GB has developed four categories of flexibility services that are designed to help balance supply and demand in the electricity grid ("Energy Networks Association (ENA," 2022). These categories are:

- Secure: This category includes services that ensure the security of supply and the stability of the electricity grid, such as frequency response and reactive power.
- Sustain: This category includes services that support the integration of renewable energy and the reduction of greenhouse gas emissions, such as demand-side management and energy storage.
- Dynamic: This category includes services that enable the dynamic management of the electricity grid, such as load balancing and voltage control.
- Restore: This category includes services that help to restore the electricity grid in the event of an interruption or disruption, such as emergency response and black start.

These categories provide a framework for the classification and development of flexibility services and help ensure that the services provided are consistent with the goal of creating a more flexible, efficient and sustainable energy system. The 4 ENA services are presented in Figure 3:3.

	Sustain	Secure	Dynamic	Restore
Use-case	Scheduled	Pre-fault	Post-fault	Post-fault network restoration
Availability Payment	Yes, for scheduled availability pre-agreed with contractior	Yes, payment for availability at week-ahead	Yes, payment for availability at week-ahead	No
Utilisation Payment	Yes	Yes	Yes	Yes
Availability Declarations	Week-ahead. By midnight every Wednesday for the following week (Mon-Sun)	Week-ahead. By midnight every Wednesday for the following week (Mon-Sun)	Week-ahead. By midnight every Wednesday for the following week (Mon-Sun)	Week-ahead. By midnight every Wednesday for the following week (Mon-Sun)
Availability Acceptance	Week-ahead. By midnight every Thursday for the following week (Mon-Sun)	Week-ahead. By midnight every Thursday for the following week (Mon-Sun)	Week-ahead. By midnight every Thursday for the following week (Mon-Sun)	Week-ahead. By midnight every Thursday for the following week (Mon-Sun)
Dispatch Notice	Fixed within contract and notice sent 15 minutes ahead of requirements	Fixed week-ahead on acceptance of availability and notice sent 15 minutes	Notice sent 15 minutes ahead of requirements	Notice sent 15 minutes ahead of requirements

Figure 3:3 Flexibility services developed by ENA Open Network's Project

For a Flexibility Provider (FP) to participate in the flexibility market, it must adhere to the protocols established by its designated DNO. The FP must respond in a prompt manner to the Periodic Indicative Notice (PIN) issued by the DNO. The PIN is an open notice that is published annually by the DNOs so that interested flexibility seller parties can register their interest. Upon receipt of the FP's offer, the DNO will proceed to issue a Pre-Qualification Questionnaire (PQQ) which includes mandatory questions pertaining to the FP's energy assets. The PQQ contains extensive queries regarding the FP's energy assets.

The DNO must evaluate the PQQ response and determine if the answers provided are sufficient to meet its requirements. If the answers are deemed unsatisfactory, the DNO must inform the FP and work towards resolving the issues (Ofgem, 2020a). If the answers are deemed satisfactory, the DNO will inform the FP and include it in the Dynamic Purchase Scheme (DPS). A DPS is a UK-wide process that provides customers a quick and flexible route to the market, effectively allowing the flexibility prosumers to trade their flexibility (Service, 2019). This process must be repeated annually, and it constitutes a procurement cycle (yet the frequency can vary depending on DNO or overall network needs).

For each procurement cycle, the FP must complete an Invitation To Tender (ITT) and sign the terms and conditions for the Constraint Management Zone (CMZ) it will be affiliated with. The ITT is a formal agreement via which the FP will be bidding for flexibility contracts during the procurement cycle. The CMZ dictates the geographical area for which the flexibility prosumer will be allowed to bid. The DNO will assess the ITT responses from all participating FPs. If the responses do not meet all the criteria, the DNO will notify the participants and offer the option to re-tender in the next procurement cycle. The criteria are based on the relationship between the flexibility needs of the CMZ and the flexibility options listed by the FPs in the PIN, ITT, PQQ and the DPS evaluation.

All FPs that meet the criteria will be eligible to bid for flexibility contracts, for one or multiple service windows. In most cases, the contract will include a fixed price for each required flexibility service. This means that in many cases the FP will only bid

with the available capacity, and with a willing to accept price (which will be fixed). On occasion, the contract will include a guide price, allowing the FPs to submit a price with each flexibility bid. The guide price presents an informative price that will have a higher chance of having the bid selected. The DNO must then evaluate all bids individually. For each accepted bid, the DNO will proceed to the build phase where an Application Programming Interface (API) is established for the FP's flexibility system, what will allow the DNO to settle the delivered flexibility. The non-accepted bids will be invited to re-tender in the subsequent procurement cycle. The whole flexibility contracting process is presented in Figure 3:4.

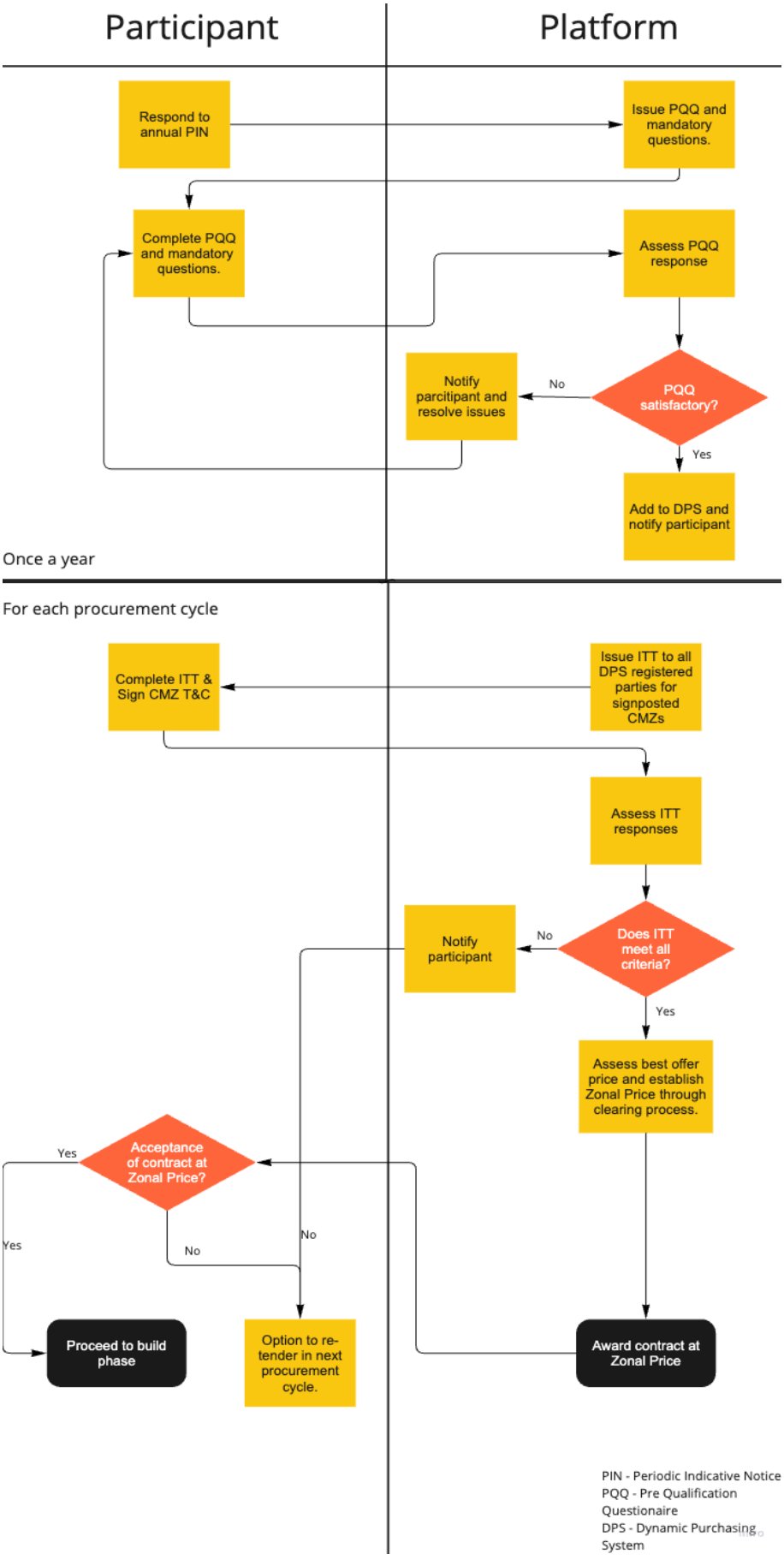


Figure 3:4 Flexibility contracting process

3.2.2 The current platforms for contracting flexibility in the UK.

In the UK, "flexibility market facilitation platforms" have emerged as intermediaries for facilitating or coordinating the trade, dispatch, or settlement of flexibility transactions between the DNO and the FP (Association, 2020). These platforms are geographically self-contained marketplaces, meaning that there is no central market platform, and act as intermediaries to the DNOs network operational performance platform, such as the advanced distribution management platform (Ofgem, 2020b). National flexibility markets, which are typically operated by the system operator, are excluded from this analysis because they are centrally coordinated and developed in-house, with limited decentralisation potential due to their single-buyer market structure and top-down governance model.

Out of the 3.7GW of distribution network flexibility tendered in the 21/22, 65% was tendered through PicloFlex, the leading independent online marketplace for trading energy flexibility in the GB (Piclo, 2022). Another platform, Flexible Power (Power, 2022), tendered roughly 20% and is a joint initiative of four DNOs with a similar function to PicloFlex. Octopus Energy has adopted a demand-reduction approach, resulting in the creation of a flexibility marketplace within its client base (“Octopus Energy,” 2021).

These platforms are self-contained marketplaces that host the evaluation of eligibility criteria and approval of FP assets and pre-qualifications (Catapult, 2020). They also store technical information about FP assets, such as their location and flexibility capacity. The platforms enhance the visibility of market participants over transaction opportunities through the proposed digital marketplace between stakeholders.

Yet, the GB has opted to allow the market to develop such platforms, without a centralised strategic direction. This has implications for the wide advancement of the flexibility markets and allows for limitations. The first identified limitation of the current platforms is the lack of transparency and standardisation across all the viable platforms. This imposes barriers for the FPs as they will have to grasp and build digital tools for the interaction with each platform. Additionally, as the platforms are not interoperable, the system operators will struggle to understand potential opposing actions at national and local flexibility markets. The current flexibility

platforms also do not have strong cybersecurity considerations, increasing the cyber risk for participants.

In this context, a DLT-based flexibility marketplace has the potential to offer a number of benefits that can address these challenges, including increased transparency, data security, process automation, and interoperability:

- Transparency and traceability: The current flexibility platforms may lack transparency and traceability in the transactions between the DNO and FP, leading to potential errors and disputes. A DLT-based platform can provide a tamper-proof ledger of transactions, ensuring transparency and traceability in the market (Agency, 2019) (Saberi et al., 2019).
- Data integrity: In the current scenario, there is a risk of data manipulation or errors in the data shared between the stakeholders, leading to potential disputes. A DLT-based platform can ensure secure data sharing and the integrity of the data shared between the stakeholders (Ferrag et al., 2018).
- Interoperability: Existing flexibility platforms may not be interoperable, leading to data silos and inefficiencies in the market. A DLT-based platform can improve interoperability between different flexibility platforms, ensuring a more efficient and unified market (Andoni et al., 2019) (Aitzhan and Svetinovic, 2018).
- Process automation: The current processes of asset pre-qualification, bidding, and contracting are prone to human and data processing errors and may be time-consuming and generally expensive. A DLT-based platform can automate these processes through the use of smart contracts, reducing the risk of errors and increasing efficiency(Silvestre et al., 2020).
- Security: The current flexibility platforms may be vulnerable to fraud and cyberattacks, leading to security concerns in the market. A DLT-based platform can provide a secure and immutable record of transactions, reducing the risk of fraud and cyberattacks in the marketplace (Mylrea and Gourisetti, 2017).

3.3 A DLT-based framework for contracting flexibility services

3.3.1 The general set-up of the DLT network

Firstly, the accessibility of the platform has to be defined. A DLT-based flexibility marketplace can be either public or private and permissioned or permissionless

(Zheng et al., 2017). A public, permissionless network allows anyone to participate in the marketplace and let's anyone act as a validator, while a private, permissioned network restricts validators to a specific group of trusted participants, also restricting the participants on the network based on an "invite only" system.

In the context of a flexibility marketplace, it is more appropriate to have a private, permissioned network, as it offers greater control over the participants and can ensure the confidentiality and security of sensitive data (W. Y. C. Wang et al., 2019). In this setup, the validators, who are responsible for maintaining the integrity of the network, are the DNOs (Guerrero et al., 2021a). For a FP to participate in the DLT network, it must first pass the DPS check ("Energy Networks Association (ENA," 2021). This is a pre-qualification process that assesses the FP's energy assets and confirms that they meet the standards set by the DNOs. Once the FP has passed the DPS check, it can join the permissioned network and start participating in the flexibility market.

The benefit of this setup is that only trusted participants are allowed to join the network, reducing the risk of fraud and ensuring a secure environment for transactions. The DNOs, as validators, in a private, permissioned network are selected based on their level of trust and technical competence, and they are responsible for maintaining the integrity of the data stored on the network. This structure provides a higher level of security, as malicious actors are less likely to compromise the network, and it also increases efficiency as the network is not cluttered with unnecessary data. Additionally, private, permissioned networks can offer faster transaction speeds, as the validators can communicate directly with each other and reach consensus more quickly. Overall, a private, permissioned DLT network is better suited to the requirements of a flexibility marketplace, as it can provide a secure and efficient platform for conducting transactions (Buterin, 2014). A private instance of the Ethereum blockchain has been forked and created to sustain the marketplace on this chapter. Ethereum has been adopted as at the time of the publication presents the most robust development functionality for the use-case, thus allowing to develop the whole framework as expressed above. Hedera hashgraph is also considered as a potential candidate, and as it uses the same development

programming language, the framework has been tested when deployed on Hedera as well.

In a DLT-based flexibility marketplace, the use of smart contracts architecture will provide a secure and transparent platform functionality for all the flexibility prosumers and DNOs to trade and settle flexibility (Hewa et al., 2021). The smart contracts will automate the trading process by encoding the rules and regulations of the marketplace into the code. This will ensure that all participants adhere to the same set of rules and regulations, providing a level playing field for everyone involved. The smart contracts will facilitate the trade of flexibility by automatically executing the terms of the agreement between the participants once the specified conditions have been met. This will eliminate the need for intermediaries and reduce the time and cost associated with traditional trading processes, enabling a more efficient and effective trading of flexibility. With the smart contract architecture, the participants will have a secure and transparent record of all their trades, making the marketplace more reliable and trustworthy. The benefits of a public, private, permissioned and permissionless network are presented in Figure 3:5.

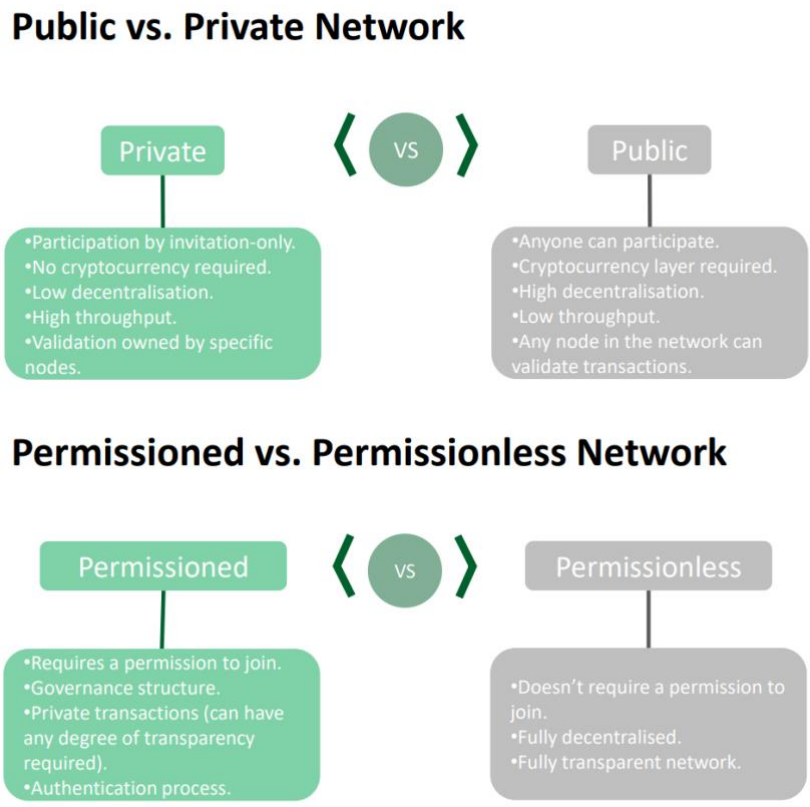


Figure 3:5 DLT public vs. private network

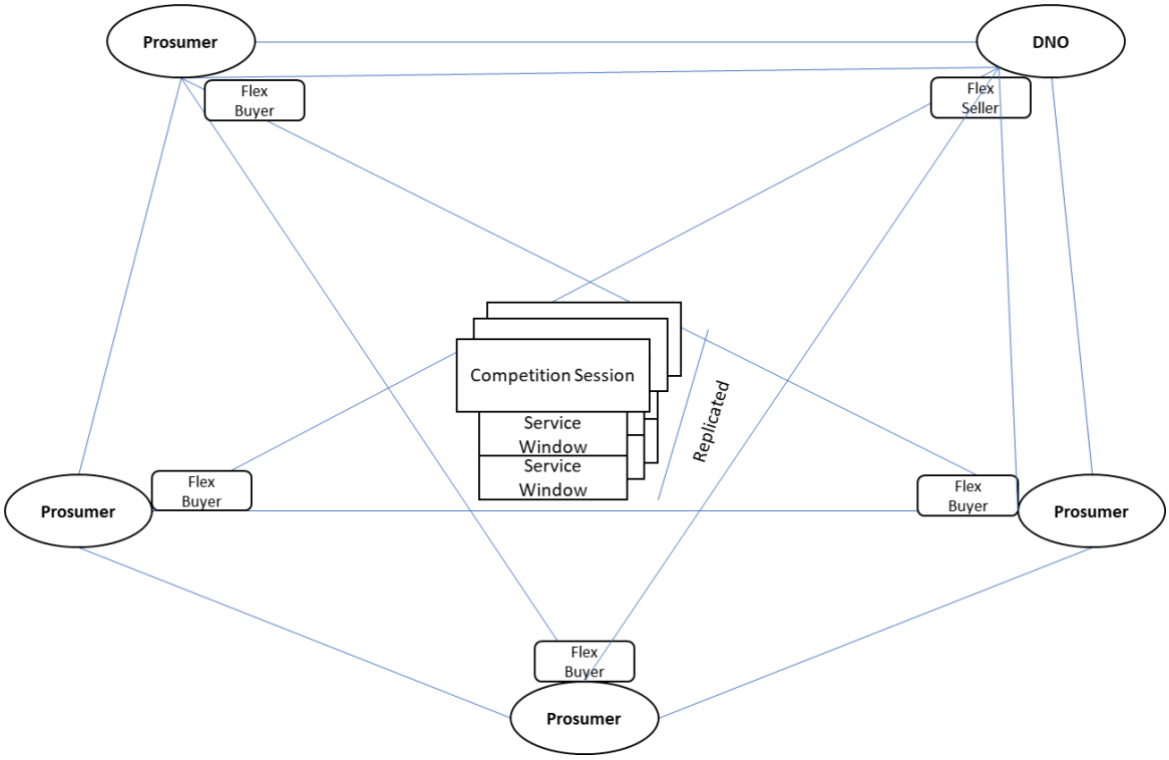


Figure 3:6 Flexibility platform network illustration

3.3.2 Decentralised flexibility contracting framework/environment

A DLT-based marketplace has been developed to facilitate the trading of flexibility between two main actors, the DNO as the buyer and the FP as the seller (Mengelkamp et al., 2018b). The marketplace operates using a smart contract type known as the Competition Contract, which serves as the marketplace platform (Kang et al., 2017).

Before any FP can participate in the marketplace, they must pass the pre-qualification criteria and be assessed by their respective DNO (Ofgem, 2020a). This process ensures that only eligible FPs are able to join the network and trade their flexibility assets. Upon successful assessment, the FP will receive a token that allows them to register to the private network and list their flexibility assets.

The marketplace operates in two distinct timeframes, the bidding process and the delivery and settlement process (Sousa et al., 2019b). This is because the bidding process normally occurs on a wholesale model in procurement cycles, while the settlement occurs at the end of the delivery period. The DNO begins the marketplace by deploying a Competition Contract for each constraint management zone and

defining the required service windows (Guerrero et al., 2021a). Each service window consists of a specific flexibility service, capacity, and timeframe. The Competition Contract encapsulates all the flexibility requirements or requests for a certain DNO in one CMZ or geographical area. This model follows the current flexibility procurement of DNOs as it incorporates all flexibility requests (i.e. the service windows) in one constraint management zone. The DNO also logs a financial escrow on the blockchain, proportional to the required flexibility services, to assure the FPs of the payment process.

With the competition contract in place, the registered FPs can submit their bids for one or multiple service windows. As part of this process, FPs must use the FlexCoin ERC20 (Ethereum Request for Comment) token to mint the required debt tokens for each service window, with the conversion rate typically set at 1 FLX = 1 kWh. The FlexCoin tokens are integral to the marketplace's security mechanism. During the delivery phase, as the FP delivers the contracted flexibility, the system reads data from the smart meter to verify the kWh of flexibility provided. For every kWh delivered, the corresponding FlexCoin tokens are burned. This burning process ensures that the tokens are only used for actual flexibility delivery and cannot be reused or repurposed fraudulently. By burning the tokens, the system guarantees that the financial transactions are directly tied to the physical delivery of energy, enhancing the integrity and trustworthiness of the marketplace.

The marketplace leverages ERC20 tokens (Foundation, 2020), specifically FlexCoin, which is a standardised digital token on the Ethereum blockchain. ERC20 is a technical standard used for smart contracts on the Ethereum blockchain that implements tokens. It defines a common list of rules that all Ethereum tokens must adhere to, allowing developers to accurately predict how new tokens will function within the larger Ethereum ecosystem. In this marketplace, the FlexCoin ERC20 token is used as the primary currency for transactions, ensuring interoperability and secure exchanges between participants. FPs mint the required FlexCoin tokens during the bidding process, which are later burned upon successful delivery of flexibility services (Khan and Salah, 2018).

The FPs are also required to lock in a financial escrow as collateral in case they fail to deliver the contracted flexibility. Once the bidding time has ended, the DNO must assess each of the submitted bids and make a selection, either individually or in combination. The DNO makes this selection "in-house", outside of the blockchain marketplace, to maintain maximum control over the bids. Upon completion of the selection process, the DNO informs the successful FPs of the outcome. This occurs on chain as the system is designed to easily allow for traceability and transparency.

In the delivery and settlement timeframe, the DNO notifies the FP to begin delivering the required flexibility, according to the timeline specified in the flexibility contract, this happening on-chain. As the flexibility is delivered, the system reads data directly from the smart meter to verify the kWh of flexibility provided. For every kWh delivered, the corresponding FlexCoin tokens are burned, ensuring that the tokens are only used for actual flexibility delivery.

The FP then delivers the agreed-upon flexibility and the DNO must validate the delivery. If the FP fails to deliver the required flexibility, the DNO retains the FP's escrow as a penalty. If the FP has successfully delivered the flexibility, the DNO unlocks the FP's escrow and sends the reward, as agreed upon in the bidding process, along with the returned escrow to the FP. At the end of the process, the final flexibility payment is sent to the FP, closing the service window. The sequence diagram is presented in Figure 3:7.

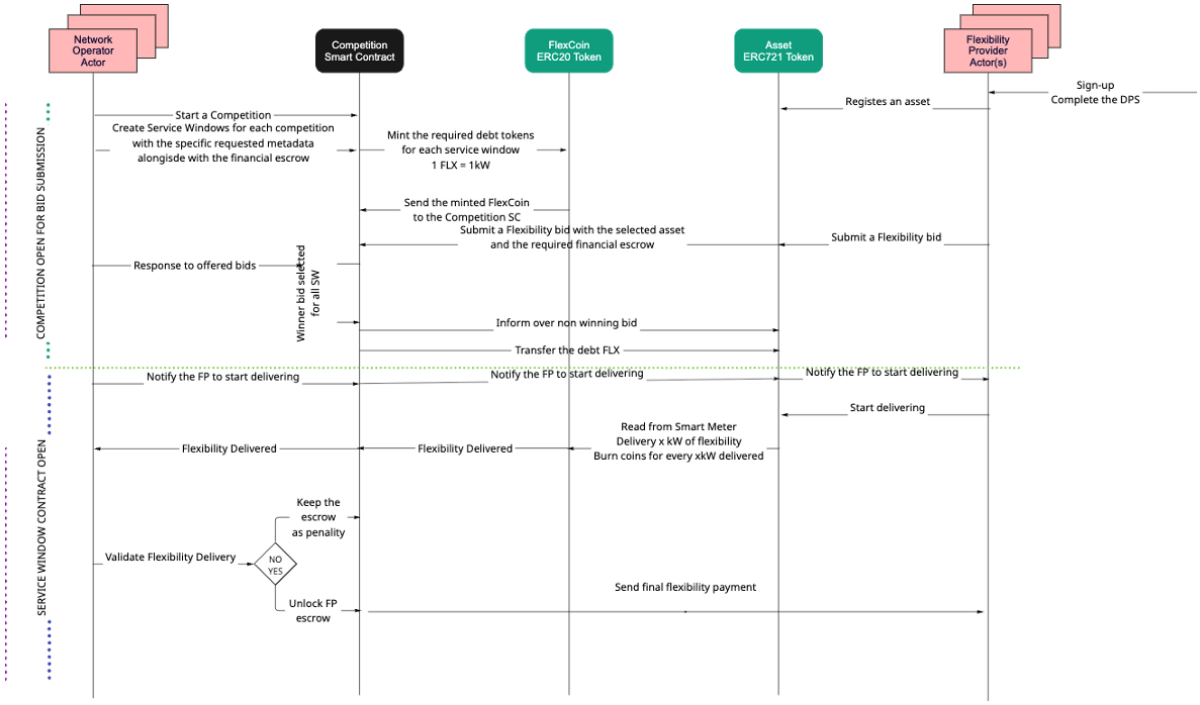


Figure 3:7 Flexibility marketplace core sequence diagram

In the design of this model, two distinct actors have been defined with specific responsibilities. The DNO actor has the duty of deploying competition smart contracts, establishing service windows, communicating the results of bidding to the FPs, and whitelisting the flexibility assets of FPs so that they can participate in the bidding process. On the other hand, the FPs are responsible for registering their flexibility assets and submitting bids for each service window they have been granted access to. This design reflects the prevailing conditions of the flexibility market in GB. It should be noted that a DNO has the capability to deploy multiple competition smart contracts and create multiple service windows for each procurement cycle. The FPs, however, can only bid for the service windows they have been whitelisted for by the DNO.

The DNO has the authority to deploy a competition smart contract for each constraint management zone where procurement of flexibility is necessary. The DNO is exclusively the owner of these smart contracts. The competition smart contract is composed of two fundamental components: data structures and functions. There are a total of five primary data structures (represented as arrays) in the smart contract, in addition to mappings (utilising bi-dimensional arrays) and events to signal specific actions within the smart contract. The functions of the competition smart contract

embody the essential processes of the competition, including deployment, bidding, market clearing, and payment.

The central data structure of the Competition is the ContractCriteria, which encompasses key information related to the address of the contract, its owner, the requested minimum capacity, and the timeframe for the competition It is worth noting that the DNO, as the owner of the Competition smart contract, has the ability to deploy multiple service windows. Each service window is designated to a specific type of flexibility service and is reflected in the ServiceWindow (SW) structure. The ServiceWindow structure details the bidding timeframe, the required capacity, the minimum run and response times, the required flexibility service, the guide availability price, the utilisation price (or a constant that denotes a fixed price), and a flag indicating the status of the service window which will dictate if the service window is opened or close. The Bid data structure outlines all the information required for a flexibility asset to submit a bid, including the address and owner of the asset, the capacity, the maximum runtime, the availability and utilisation price, the period of the SW, and the SW id. The last two data structures are two enumeration data types, designed for the purpose of reducing data stored on the smart contract for efficiency purposes. The first enumeration reflects the type of flexibility service used for the service window, while the second enumeration signifies the status of the service window, which can be ongoing, delivered, or failed.

The mapping data structure has been implemented to establish links between various IDs and their respective addresses in the distributed ledger. A ServiceWindow mapping has been established to associate the ID of a service window with its populated ServiceWindow structure. A bidding mapping has also been created to link the ID of a bid with its corresponding Bid data structure. Two additional mappings have been created to connect submitted and winning bids with their relevant service window within a competition.

The events data types have been established to indicate critical actions in the smart contract. These events allow the functions to execute specific actions when triggered, for instance, to cease accepting bids for a service window when the

bidding period has ended or to indicate the start time for a service window for flexibility prosumers.

The functions implemented in the smart contract are aligned with the market functionality. The functions have been designed to automatically validate bid submissions, voltage levels, FP capacity, and payments.

The smart contract used within the DLT-based marketplace has been meticulously designed to handle the entire lifecycle of flexibility procurement—from creating service windows to bidding, delivery, and final settlement. To facilitate this, several key functions and mappings have been integrated into the smart contract to ensure efficient and transparent management of the process.

Mappings in this smart contract serve as essential data structures that link unique identifiers to their respective data entities. There are several important mappings utilised in this contract:

- mapping(uint256 => ServiceWindow) public idServiceWindows -This mapping associates a unique identifier (`uint256`) with a `ServiceWindow` structure. Each `ServiceWindow` represents a specific flexibility service offering, detailing the required capacity, timeframe, and other relevant criteria. By utilising this mapping, the smart contract can quickly retrieve all necessary details about a particular service window using its ID.
- mapping(uint256 => Bid) public bidIds - This mapping links a unique bid ID (`uint256`) to a `Bid` structure. The `Bid` structure contains all the details of a bid submitted by a FP, such as the capacity offered, price points, and the specific service window for which the bid is made. This allows the contract to keep track of all bids and their associated data.
- mapping(uint256 => uint256[]) public bidsForWindow - This mapping associates each service window ID (`uint256`) with an array of bid IDs (`uint256[]`). Essentially, this creates a list of all bids that have been submitted for a particular service window, enabling the DNO to evaluate and compare bids effectively.

- mapping(uint256 => uint256) public bidsWonWindow - This mapping is used to link a service window ID with the bid ID that won the competition for that particular window. By storing the winning bid, the smart contract can easily track which FP has been awarded the contract for each service window.

Events in the smart contract are designed to emit logs that can be captured by external systems, providing transparency and traceability for various actions. The following events are crucial for the operation of the marketplace:

- event ServiceWindowCreated(uint256 serviceWindowId) - This event is emitted when a new service window is added to the competition. It signifies that a new opportunity for flexibility procurement has been created and is now available for bidding.
- event BiddingOn() - This event indicates that the bidding process is now open for a specific service window. FPs can begin submitting their bids at this stage.
- event Deposit(address payer, uint256 amount) - Emitted when a financial deposit is made into the escrow by a FP. This deposit acts as collateral to ensure that the provider fulfills their obligations if they win the bid.
- event Withdraw(address payer, uint256 amount) - This event signifies that the bidding period has ended, and any unselected bids may have their deposits withdrawn. It also marks the conclusion of the escrow process for bids that were not successful.
- event WindowStart(address meterID) - Emitted when the flexibility delivery window begins. This event is tied to the start of the service delivery period, signaling that the FP should commence delivery as per the contract.
- event WindowWinner(address meterID) - This event identifies the winner of a service window, linking the winning bid to the FP’s meter ID. This is crucial for tracking the successful execution of the flexibility contract.
- event WindowEnd(address meterID) - Emitted when the service window ends, this event marks the completion of the flexibility delivery period. It signals that the delivery has been completed, and the final settlement process can commence.

The functions implemented in the smart contract are directly aligned with these mappings and events, providing automated processes that govern the entire flexibility procurement lifecycle:

- Bidding Process: The smart contract automatically opens the bidding process through the `BiddingOn()` function, enabling registered FPs to submit their bids. The system tracks all bids and associates them with their respective service windows using the `bidsForWindow` mapping.
- Deposit and Escrow: The `Deposit()` function handles the financial deposits from the FPs, which are stored in escrow until the bidding concludes. This ensures that only serious and capable bidders participate in the marketplace.
- Service Window Management: The `ServiceWindowCreated()` and `WindowStart()` functions manage the lifecycle of service windows—from creation to the start of delivery. These functions are integral in maintaining the integrity and timing of the marketplace operations.
- Bid Selection and Settlement: Once the bidding period ends, the DNO evaluates the bids, and the `WindowWinner()` function is called to record the winning bid. After the delivery period concludes, the `WindowEnd()` function triggers the final settlement, where the escrow funds are released, and the agreed payments are made to the FP.

3.3.3 The market clearing algorithm

The implementation of a market-clearing algorithm for analysing the feasibility of using DLT for on-chain computation in a flexibility marketplace has been conducted. It is assumed that the DNO will allow the competition smart-contract to automatically clear all the flexibility bids for each service window (Teotia et al., 2021). The market-clearing algorithm has been based on the principles of price priority, where the FPs must provide a bidding price for each bid, and no guide price will be issued by the DNO.

A single-auction market design has been adopted, with a single buyer (DNO) and multiple sellers (FPs) (Y. Wang et al., 2019). The sellers can submit quoted prices at any time during the transaction timeframe, and at the market clearing time, the smart contract will sort the sellers' prices from low to high. In the case of the same price,

the bids will be sorted according to the time of submission. The FPs are designed to always maximise their available capacity for higher profits. At the market clearing time, the required quantity of flexibility will be filled based on the offered quantity of flexibility, starting from the lowest price and moving to the highest price.

The flowchart of the entire system includes two layers: initialisation and trading. In the initialisation layer, the FPs register their flexibility assets, while the DNO registers its competitions and service windows (Christidis and Devetsikiotis, 2016b). In the trading layer, the FPs continuously send flexibility offers until the market clearing time, and the market is cleared, and the FPs are informed of the outcomes.

The transaction process, similar to (Goranovic et al., 2017) and presented in Figure 3:8 and Figure 3:9 is summarised in five steps, where “t” is the time of market clearing:

- Initialisation: DNO creates competitions and service windows; FPs register their flexibility assets.
- t-24 hours: DNO sends the final flexibility requirements for each service window in terms of quantity.
- t-30 minutes: FPs continuously submit flexibility bids in terms of their available flexibility quantity and price.
- t-15 minutes: smart-contract automatically clears the market; DNO is informed of selected bids and asked to accept or deny selection. If approved, the smart contract generates final flexibility transactions and informs the FPs of bid acceptance. If denied, smart contract informs the DNO, selects the second best and requires the DNO to accept.
- t+15 minutes: market clearing is finalised; FPs exit the auction and wait for the next one.

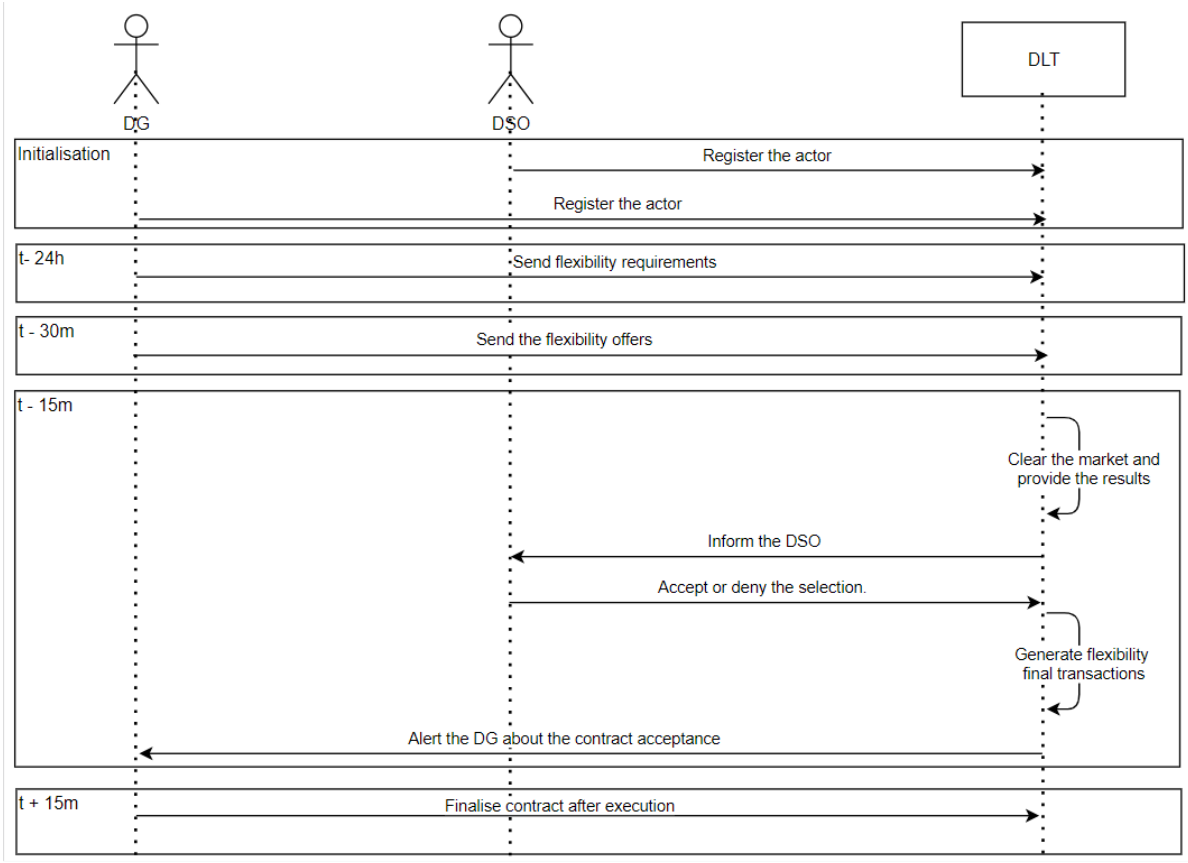


Figure 3:8 Market clearing algorithm sequence diagram

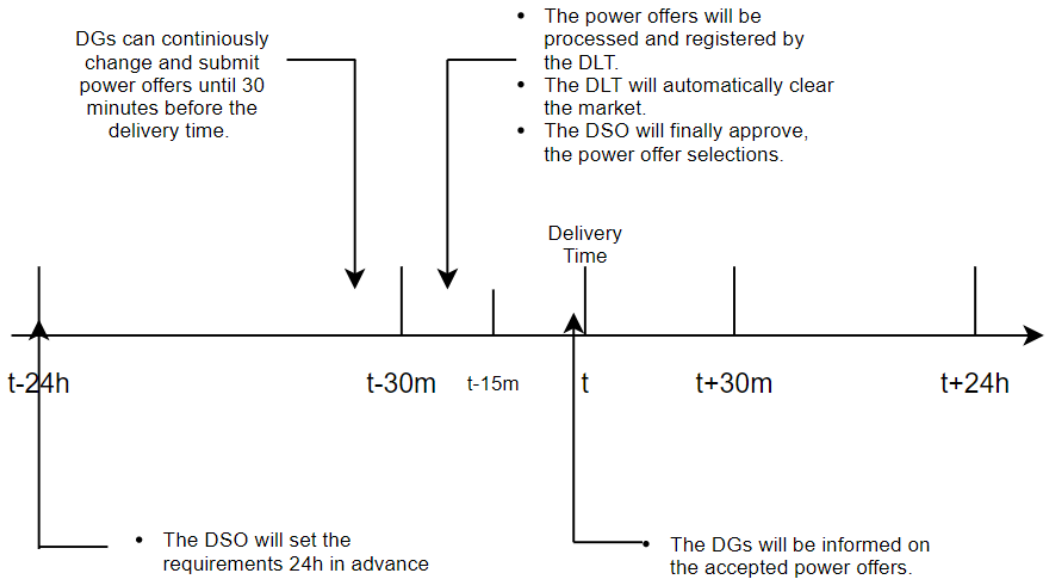


Figure 3:9 Market clearing algorithm time action diagram

3.4 Test-case and results

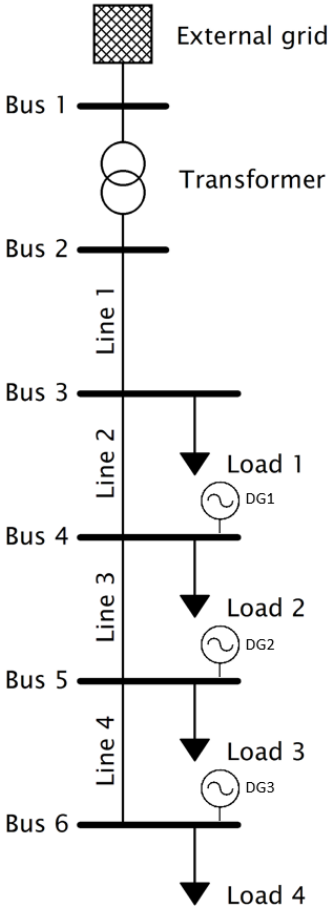
3.4.1 Test-case set up

A six-bus network has been considered as part of an independent microgrid system, as illustrated by the one-line diagram in Figure 3:10. This diagram shows the network topology, including bus interconnections, DGs, loads, and the transformer that links the microgrid to the external grid between Buses 1 and 2. This transformer is used for voltage regulation.

The electrical characteristics of the buses, such as their starting voltages, and the details of the connected loads, including active and reactive power, are not depicted in the diagram but are instead provided in Table 1 and Table 2 (attached to Figure 3:10), respectively.

A power flow analysis conducted using IPSA revealed voltage violations at each bus. IPSA was selected for this study because it is a widely used, industry-standard software tool for modelling and analysing distribution networks. It offers support for unbalanced LV network modelling, handles multiple load and generation scenarios, and provides detailed voltage, current, and power flow outputs essential for flexibility studies. Additionally, IPSA offers a clear and intuitive visual interface, which facilitated the modelling process and enhanced the efficiency of the analysis. This was particularly beneficial given that power flow analysis is not the primary focus of this chapter. It is assumed that the DNO is aware of these issues and issues flexibility capacity requirements every 30 minutes to address them.

All three FPs, each associated with a DG, are assumed to be eligible to participate in the flexibility market. The DNO's flexibility requirements over time are presented in Figure 3.11, while Figure 3.12 outlines the available capacity and pricing for each DG's flexibility offerings for each 30-minute interval.



Bus Name	Bus Starting Voltage (kV)
Bus 1	11
Bus 2	0.4
Bus 3	0.4
Bus 4	0.4
Bus 5	0.4
Bus 6	0.4

Table 1: Bus Starting Voltage

Load Name	Connected Bus	Active Power	Reactive Power
Load 1	2	0.02	0.01
Load 2	3	0.02	0.01
Load 3	4	0.02	0.01
Load 4	5	0.02	0.01

Table 2: Test Network Loads

Figure 3:10 Test-case network configuration

In Figure 3:11, the assumptions for this test case are adapted from ENA's 2023 Flexibility Figures and are intended to represent a simplified example of the network capacity a DNO may need to procure to avoid physical network reinforcement at an 11 kV substation. Figure 3:11 displays the time-varying flexibility demand profile from the DNO, represented as a curve with peaks during the early evening period—highlighting the typical residential evening demand spike. Figure 3:12 then compares this required flexibility against the available supply from the three distributed generators. The stacked bar format indicates the hourly contribution from each DG, while the blue line overlays the DNO's power requirement. This visualisation showcases how the aggregated DG output aligns with, or falls short of, the DNO's flexibility need over the course of the day. The interaction between Figures 3:11 and 3:12 helps contextualise the decision-making process in dispatching available flexibility while ensuring secure system operation for the distribution network.

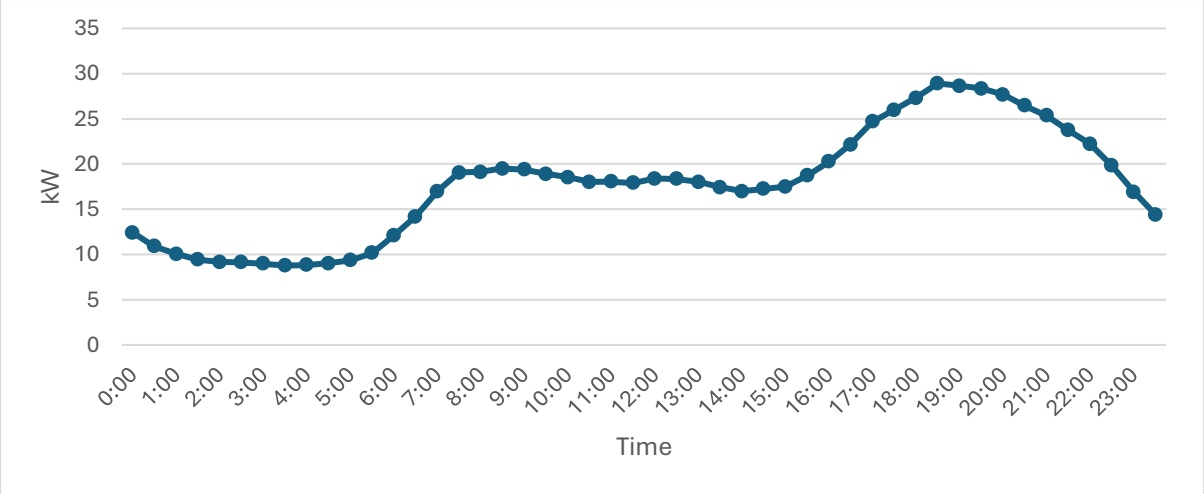


Figure 3:11 DSO flexibility requirements

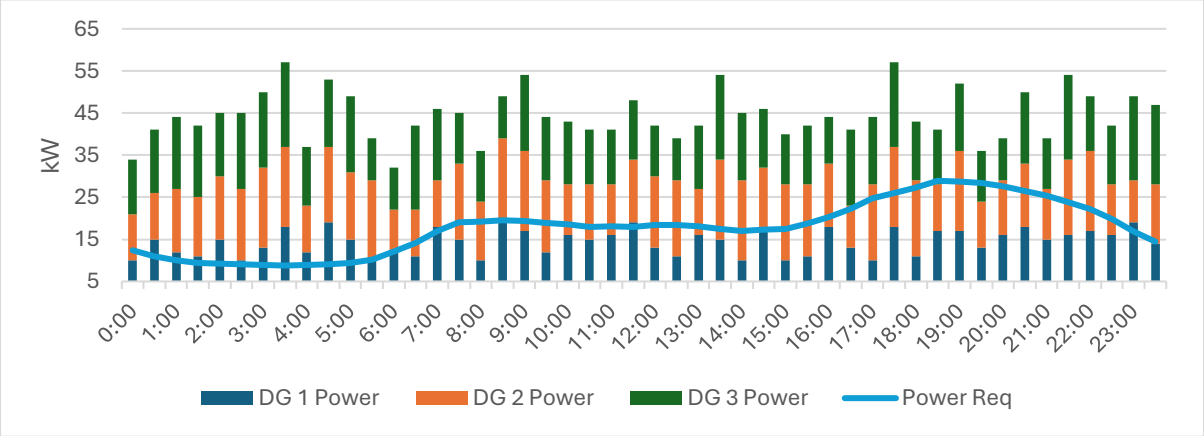


Figure 3:12 Power offers

3.4.2 Market clearing results

The marketplace algorithm sorts the bids from the FPs based on their price and the time they were submitted. The market clearing algorithm then calculates the required amount of flexibility from each FP incrementally based on price. The algorithm checks for network violations by conducting a power flow analysis. If there are no violations, the algorithm sends the results back to the DNO. A snapshot of the process for one time frame is available in figure in Figure 3:13.

Figure 3:13 Algorithm market clearing results snapshot

Figure 3:14 presents a console output with the voltage magnitude each busbar, alongside with the amount of cleared flexibility, price and address for each of the three DG.

```
([13, 11, 10], [8, 10, 10], ['0x1C4971126d1855dafE85Ba2f5b0f9ebe7475437B',
'0xabD1425BDE57259A041331F0809dF9A7E64dE4d6',
'0x92Ea930c3412bF037035a0DC532f94f3403a98d1'])
[12, 0, 0]
0    1.000000
1    1.003093
2    1.000712
3    1.000000
4    1.000000
5    1.000000
Name: vm_pu, dtype: float64
```

Figure 3:14 Console output

Figure 3:15 presents the cleared flexibility for each timeframe, and Figure 3:16 presents the voltage magnitude values, for each 30 minutes timeframe, for 24 hours.

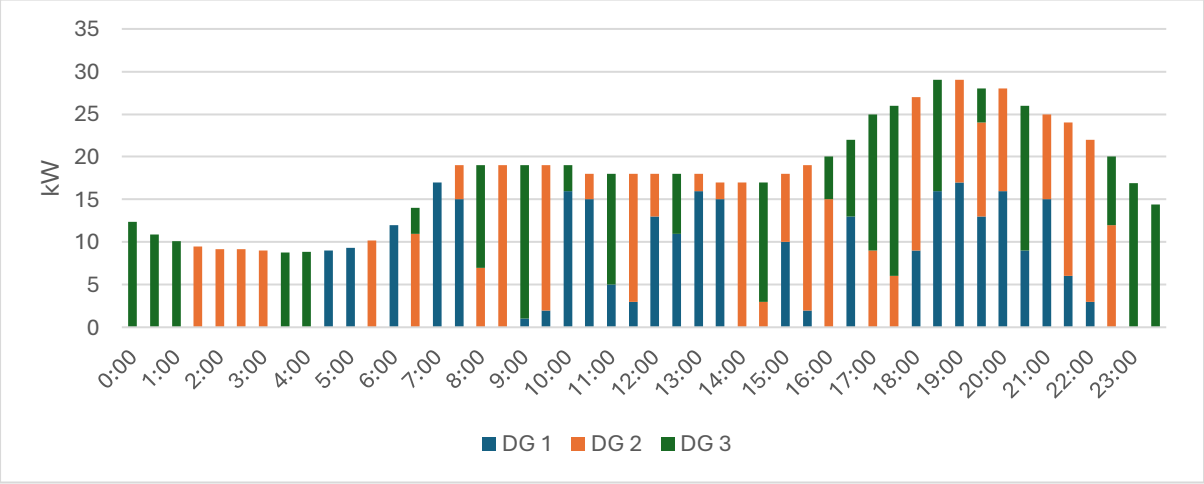


Figure 3:15 Cleared flexibility

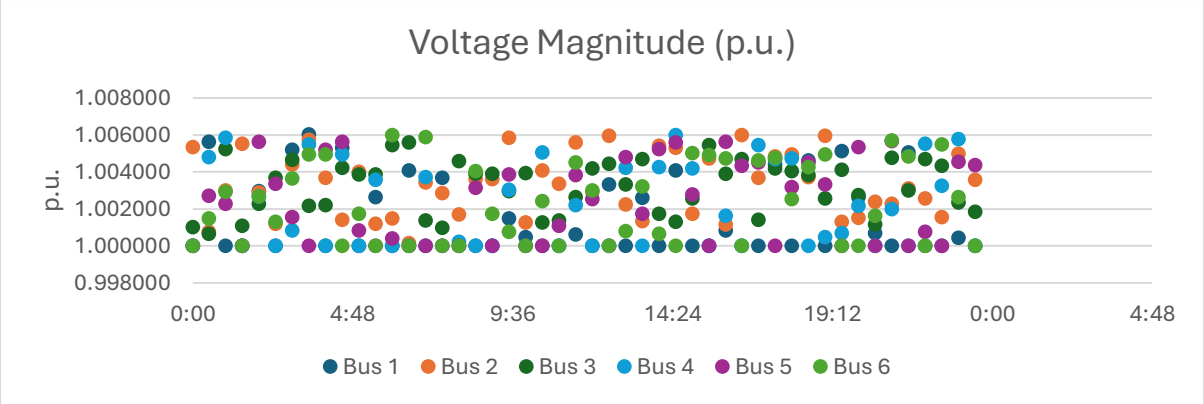


Figure 3:16 Voltage magnitude

Additionally, the escrow mechanism sends the funds from one party to the other. At the moment of initialising the flexibility contract, the tokens sit within the DNO deposit, during the delivery they sit within the FP deposit, and after the flexibility has been settled, they are burnt and while the money is sent to the FP. This mechanism proves that all the required flexibility has been delivered by the required DGs. This has been shown for a sample of 3 contracts in Figure 3:17.

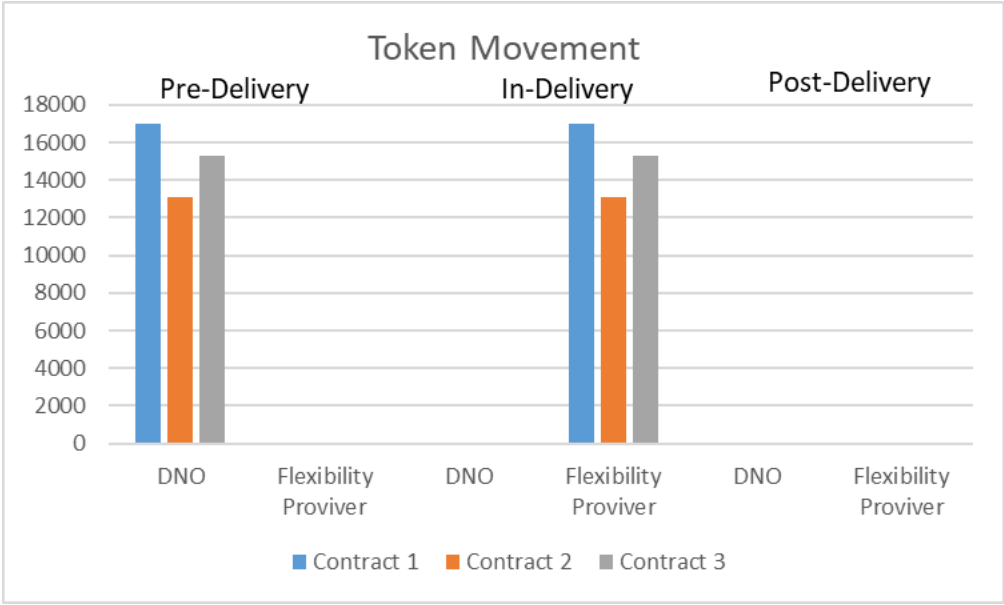


Figure 3:17 Burning token mechanism, token movement

3.4.3 DLT performance

The feasibility of using a distributed ledger architecture on the presented use-case has been investigated. The selected DLTs have been the Ethereum blockchain and

the Hedera Hasgraph. The Table 3:1 presents the key performance indicators of each DLT with the associated considerations.

Metric Name	Ethereum	Hedera	Comment
Average Throughput	15+ TPS (but this is likely to change for Ethereum 2.0.)	10,000+ (this is advertised on their website) 74.9 (last month)	The average throughput of ETH has reached the maximum with the current consensus mechanism, yet this is subject to change with Ethereum 2.0. The Hedera throughput comes from their website, therefore it is probably lower in a real case scenario.
Average Latency	13 seconds	3 seconds	In Hedera, this is the average latency for the current 13 validation nodes. Due to the gossip about gossip protocol, this might slightly increase in the future.
Average Transaction Fee	13 USD	0.05 USD	The average Tx fee of ETH has increased with a factor of 10 over the last year, due to the popularity of the DLT.
Maximum Smart Contract Size	24KB	1MB	
Maximum Gas Limit	10,000,000 to 12,500,000	300 000	Reflects the required fee to deploy a transaction on the DLT. Present the most important factor for the final transaction fee.
Average Finality Time	10-20 seconds	5 seconds	Clear indicator of the superiority of the architecture and consensus of Hedera.
Transactions Per Day	1.25 mil	1.5 mil	The TPD value of Hedera has been taken from their website. With the much higher popularity of ETH, the value doesn't seem to reflect reality

Table 3:1 DLT key performance indicators

To test the throughput and financial viability of the model, three types of computational activities have been considered. Information about these is available in the Table 3:2.

Metric Transaction Name	Computational Toughness Level	Description
Clear the market	High	This type of transaction requires high computational power from the DLT due to the amount of calculation the DLT has to perform for achieving finality. Within this type of transaction, the DLT will have to sort the power offers in terms of price, fill for the power requirements and send the results.
Send the power bids (or flexibility requirements)	Medium	This type of transaction requires medium computational power as it's a classic write type of transaction. The DLT will have to receive the information and pass it to the smart contract.
Ask for power bids (or flexibility requirements)	Low	This type of transaction requires low computational power as it's a simple read type of transaction. The DLT will have to only pass the information out of the smart-contract.

Table 3:2 DLT computational activities toughness levels

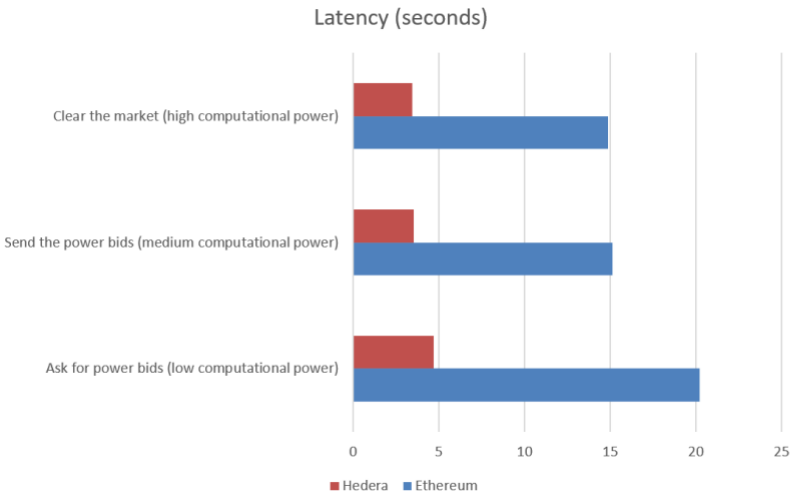
The comparison of the performance of Hedera Hashgraph and Ethereum blockchain in terms of computational activity shows that Hedera Hashgraph performs

significantly better across all three tested transactions. As illustrated in Figure 3:18, the latency for low computational power tasks such as “ask for power bids” is 20.21 seconds on Ethereum, compared to just 4.7 seconds on Hedera. For medium computational tasks like “send the power bids,” Ethereum reports a latency of 15.13 seconds, whereas Hedera completes the task in only 3.52 seconds. Even in high computational tasks such as “clearing the market,” Hedera again outperforms with a latency of 3.46 seconds compared to Ethereum’s 14.87 seconds. These differences stem from the faster consensus mechanism used in Hedera, which enables quicker validation of transactions.

However, this speed comes at the cost of decentralisation. Hedera’s consensus mechanism relies on a fixed number of permissioned nodes, making it less decentralised than Ethereum’s PoW model. Ethereum involves a larger pool of validators, increasing decentralisation and trust assumptions, but also resulting in slower transaction processing due to the computational overhead of PoW.

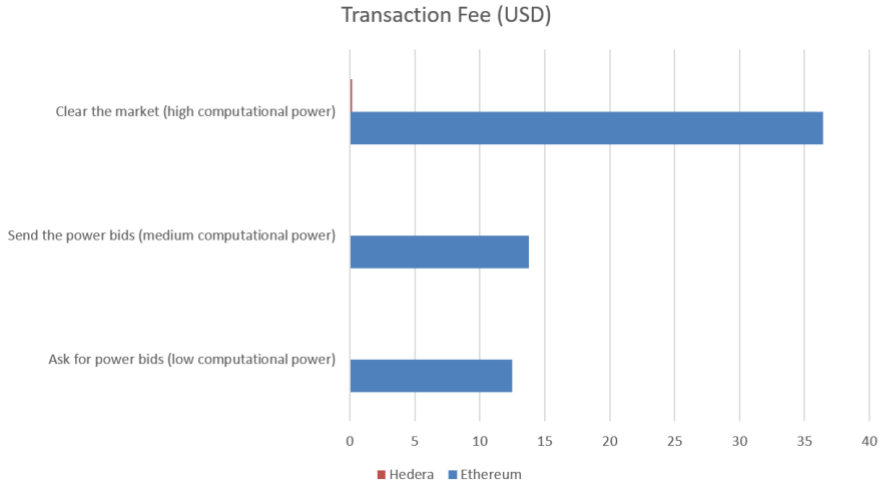
Transaction costs also highlight a stark difference in efficiency. As shown in Figure 3:19, Hedera’s fees remain very low across all transactions. For example, clearing the market on Ethereum incurs a cost of $35.6, while on Hedera, the same task costs just $0.14. Similarly, “ask for power bids” costs $12.48 on Ethereum compared to only $0.048 on Hedera. This significant difference is largely due to Ethereum’s PoW mechanism, which consumes more resources and hence drives up gas fees, especially under network congestion.

While Hedera demonstrates superior latency, as shown in Figure 3:18, and cost-effectiveness in this controlled test case, Ethereum’s ecosystem maturity, broader tool support, and high decentralisation may still make it more suitable for robust and mission-critical decentralised applications.



Transaction	Latency (seconds) Ethereum	Hedera
Ask for power bids (low computational power)	20.21	4.7
Send the power bids (medium computational power)	15.136	3.52
Clear the market (high computational power)	14.878	3.46

Figure 3:18 Latency results



Transaction	Transaction Fee (USD) Ethereum	Hedera
Ask for power bids (low computational power)	12.48	0.048
Send the power bids (medium computational power)	13.78	0.053
Clear the market (high computational power)	36.4	0.14

Figure 3:19 Transaction fee results

3.5 Conclusion

This research has showcased the viability and effectiveness of DLT in revolutionising the contracting process for flexibility services within the UK's electricity distribution networks. The implementation of DLT addresses critical issues such as the need for increased transparency, enhanced security, and improved efficiency in these systems. The findings indicate a significant potential for DLT to contribute to the UK's ambitious Net Zero targets by enabling a more sustainable and resilient energy infrastructure.

Finally, a comprehensive assessment of the long-term economic and environmental impacts of the DLT implementation is necessary. This involves evaluating the framework's contribution to sustainable energy goals over extended periods and its

influence on policy and strategic decision-making. Understanding these impacts will provide a roadmap for future enhancements and will be instrumental in guiding policy decisions for sustainable energy development.

4 A zero-knowledge proof mechanism for flexibility delivery settlement

This chapter addresses privacy and security challenges in flexibility settlement by introducing a ZKP-based mechanism. To develop the proof, a Perdesen Commitment scheme has been adopted which is used to cryptographically commit to baseline and actual energy profiles without revealing sensitive data. The scheme inherits the zk-SNARKs standard, which enables succinct proofs of flexibility delivery, verified without exposing underlying data. The MINA blockchain has been used as the DLT platform and architecture for the development of the research due to its excursive zk-SNARK capabilities, ensuring efficient and scalable verification. Furtherly, the work explores a larger integration of the algorithm with smart meters, decentralised storage, and blockchain to automate settlement while preserving privacy. A realistic use-case has been developed to test the algorithm, with 6 flexibility assets connected under the same LV substation. A standard baseline for the EV and PV profile of each asset has been adopted, alongside with a classic wintertime actual delivery profile of each asset. The results indicate the algorithm automatically proofs the delivery of flexibility, without the need of a third settlement party.

4.1 Introduction

The global transition towards a more sustainable and decentralised energy system has introduced new challenges and opportunities in the management and operation of electricity networks. Flexibility has become a crucial element in facilitating this transition (Papavasiliou and Oren, 2018). The integration of flexibility services allows for the increased adoption of renewable energy sources, enhances grid stability, and enables a more efficient balance between supply and demand (Zhao et al., 2020).

However, the current process of delivering and settling flexibility services presents several challenges. The existing system involves intricate data exchanges, raises concerns regarding privacy and security, and can be administratively cumbersome for all involved parties (Mohandes et al., 2019). This chapter aims to investigate these challenges in depth and explore the potential of a ZKP approach in revolutionising the flexibility delivery settlement process. The research focuses on how ZKPs can offer enhanced security, privacy, and efficiency for all stakeholders in the flexibility market, thereby addressing the identified challenges and facilitating the smooth integration of flexibility services into the evolving energy landscape.

The flexibility delivery settlement process is a crucial part of the flexibility market, ensuring that services are delivered as contracted and that payments are made accurately. This process involves several steps:

- Delivery of Services: Once a flexibility contract is awarded, the provider is responsible for delivering the agreed-upon services. This could involve adjusting their energy production, consumption, or storage in response to signals from the DNO. The provider's performance is monitored and recorded, often through automated systems, to verify that the services are delivered as contracted (“Energy Networks Association (ENA,” 2021).
- Verification: After the delivery of services, the DNO verifies the performance of the provider against the contract terms (Ofgem, 2017). This involves comparing the actual performance data (e.g., the amount of energy produced, consumed, or stored) with the contracted amounts. This verification process is crucial to ensure that providers are meeting their obligations and that the network's needs are being met.
- Calculation of Payments: Once the provider's performance has been verified, the DNO calculates the payment due to the provider. This is typically based on the price agreed in the contract and the actual amount of flexibility provided (E.L.E.X.O.N., 2021a). For example, if a provider contracted to reduce their energy consumption by a certain amount during a peak demand period, they would be paid based on the actual amount of consumption they reduced. In other terms, this thesis chapter only focuses on the calculation for the utilisation price, not availability.

- Settlement: The final step in the process is the settlement, where the DNO makes the payment to the provider. This involves transferring funds from the DNO to the provider, often through a secure electronic payment system. The provider's performance and the payment are then recorded for future reference and auditing purposes (E.L.E.X.O.N., 2021a).
- Dispute Resolution: If there is a disagreement between the DNO and the provider about the delivery of services or the payment, a dispute resolution process may be initiated. This could involve reviewing the performance data, the contract terms, and any other relevant information to resolve the dispute (Advice, 2019).

4.2 Technical Background

4.2.1 Challenges with the current flexibility delivery settlement process

In the current flexibility market, DNOs require FPs to develop an application programming interface (API) that links their energy asset's metering device(s) (Agency, 2019). The metering devices, which are typically smart meters, are owned by the energy suppliers ("ENA Open Networks Project," 2021). However, the data generated by these devices is owned by the prosumers (FPs) (GB, 2021). The developed by the FPs allows the DNOs to access the necessary data for calculating the baseline energy profile and settling the flexibility services (Company, 2019). In the current process, the data from the smart meters is collected by the Data Communications Company (DCC) (Company, 2019) , and then Elexon, the entity responsible for the balancing and settlement of the UK's electricity market, alongside the DNO uses this data for settlement purposes.

The FPs will be responsible for developing this API, and they will be provided with a tutorial that explains the architecture, data requirements, and other necessary information to assist them in the development process (E.L.E.X.O.N., 2020). This API is crucial for extracting data before the delivery of flexibility services. The data required typically includes the energy asset's load or generation profile, expressed in Watts over a specific period. Using this data, the DNO calculates a baseline energy profile ("ENA Open Networks Project," 2021). During the delivery of flexibility services, the DNO compares this baseline profile with the current profile to determine the amount of flexibility provided (ESO, 2020).

However, this existing flexibility delivery settlement process raises several security and privacy concerns for both FPs and DNOs. From the perspective of FPs, the requirement to construct an API (Application Programable Interface) from their energy asset's metering device(s) to extract data for baseline energy profile calculation introduces potential vulnerabilities (Cybersecurity, 2019). The act of sharing the entire energy profile for baselining may unintentionally expose sensitive information. This information could include energy consumption patterns and asset capabilities, which could make their systems susceptible to unauthorised access and potential cyber threats. A summary of the most relevant cybersecurity issues is presented in Table 4:1.

Identified Problem	Affected Actors
Cybersecurity concerns in a high-volume transaction environment.	DNO, FP, Regulator
Data sensitivity, consumer data is at risk.	FP, Regulator.
Locked-in data and lack of transparency over flexibility transactions.	DNO, FP, Regulator
Lack of standardised data formats and solution for all customers.	DNO, Regulator
Increased market barrier for flexibility prosumers.	FP, Regulator

Table 4:1 Cybersecurity issues

The verification of flexibility delivery, which typically involves comparing the baseline energy profile with the current energy profile during delivery, could also potentially reveal sensitive data and energy profiles of FPs. This is particularly concerning for small-scale prosumers who may have limited resources to invest in robust data management systems that ensure security and protect against data breaches (Guerrero et al., 2021b).

DNOs also face security and privacy risks in the collection and processing of flexibility data. The task of handling large volumes of data increases the risk of cyberattacks and unauthorised access to sensitive information (E.N.I.S.A., 2019). This information could include energy consumption patterns and customer data, which are both valuable and sensitive. Therefore, robust security measures are

necessary to protect the data, maintain privacy, and ensure the integrity of the flexibility market. In the flexibility market, a vast number of transactions occur, involving the exchange of significant amounts of data. This high-volume transaction environment can pose cybersecurity concerns for DNOs, as it increases the potential points of vulnerability that could be exploited by malicious actors. For FPs (FPs), this environment can raise concerns about the security and privacy of their data, as they may be required to share sensitive information about their energy assets and consumption patterns.

Moreover, the regulatory body, Ofgem, which is responsible for overseeing the electricity market, may face challenges in monitoring and regulating this high-volume transaction environment (Centre, 2020). While Ofgem does not directly receive the data exchanged between DNOs and FPs, they require visibility over the transactions to ensure the market's integrity and protect consumers' interests. The lack of such visibility can hinder Ofgem's ability to effectively perform its regulatory duties (Ofgem, 2020c). Ensuring the privacy and security of consumer data is crucial to maintain trust in the flexibility market and comply with data protection regulations.

Both FPs and Ofgem, which is responsible for protecting consumers in the energy market, are affected by this issue. In the future, when flexibility will become more and more utilised, Ofgem will require a greater visibility in the settlement process to ensure and audit fair settlements and data handling to protect the prosumers and the market.

The current flexibility market can lead to situations where data is locked in, and there is a lack of transparency over flexibility transactions (Office, 2018). Locked-in data, in this context, refers to the fact that none of the three parties have full visibility over how the data is utilised and processed towards settlement. This affects DNOs, FPs, and Ofgem. Without transparency, it can be challenging to verify that services are delivered as contracted and that payments are made accurately. It can also make it harder for potential new entrants to understand the market, potentially limiting competition and innovation (Taskforce, 2019). Therefore, the flexibility prosumer might not want to engage in flexibility markets due to cybersecurity concerns, which will pose difficulties to the DNOs towards filling all the flexibility requests. In such

situation, Ofgem will have then re-design the regulatory environment for data sharing as the risk that the demand will be much higher than the supply could lead to imbalanced and not efficient markets.

The absence of standardised data formats in the flexibility market can lead to inefficiencies and difficulties in managing and analysing data. This primarily affects DNOs and Ofgem. Without standardisation, each participant, including DNOs and FPs, may use different data formats, making it harder to share and compare data (Energy, 2018). This lack of standardisation can lead to inefficiencies and potential errors in the data exchange process, as each party may need to manually convert or interpret the data received from others.

For DNOs, the lack of standardised data formats can make it more challenging to aggregate and analyze data from multiple FPs, hindering their ability to effectively manage the grid and plan for future flexibility needs. FPs, on the other hand, may face difficulties in ensuring that their data is compatible with the DNO's systems, leading to potential delays or errors in the settlement process.

Furthermore, the lack of standardisation can make it harder for Ofgem to monitor and regulate the flexibility market effectively. Without a common data format, Ofgem may struggle to compare and analyze data from different participants, making it more challenging to identify potential issues or irregularities in the market.

This can slow down transactions by the fact that the data format will have to be updated by the receiving entity (data consumer), increase the risk of errors, and make it more difficult to monitor and regulate the market.

The current process for delivering and settling flexibility services can increase market barriers for flexibility prosumers. This affects both FPs and Ofgem. The requirement to develop APIs and share extensive data can deter potential FPs from participating in the market. This can limit the number of participants in the market, reducing competition and potentially slowing down the transition to a more flexible and sustainable energy system.

In an ideal scenario, FPs should only be required to share the delivered flexibility amount, measured in kWh of either generation or consumption, without divulging their energy asset's energy profile. This approach would ensure that critical data is communicated to the DNO while safeguarding sensitive asset information. It would strike a balance between the operational needs of the flexibility market and the privacy and security concerns of the market participants, as presented in Figure 4:1.

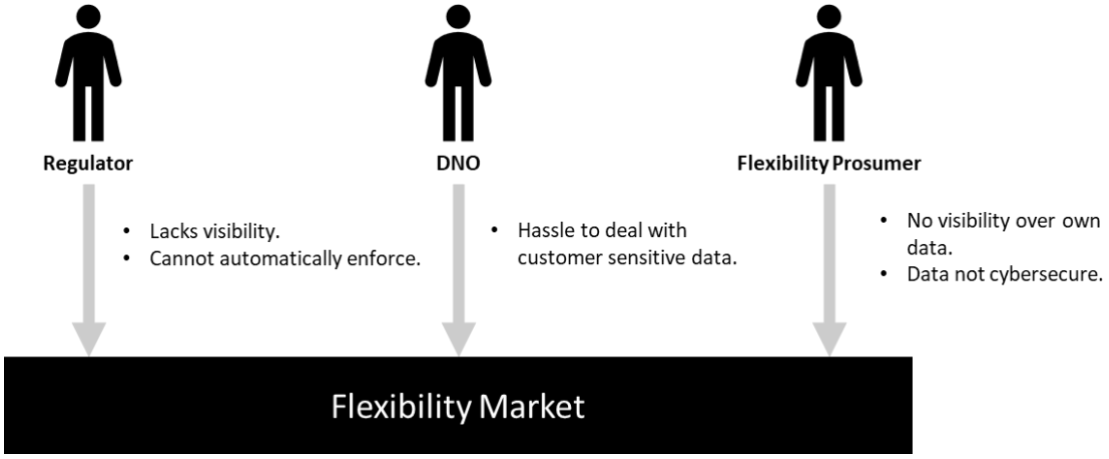


Figure 4:1 Actor interaction

4.2.2 Data requirements for flexibility delivery settlement process

Data plays a critical role in the settlement of flexibility within the UK's energy market (Mohandes et al., 2019). The ability to accurately record, verify, and settle flexibility transactions is heavily reliant on comprehensive and reliable data. Such data needs to extend beyond the mere quantity of flexibility delivered and the timeframe, as it forms the basis for various calculations and estimations required to enable smooth transactions in the flexibility market. In a broader sense, the necessary data encapsulates demand, generation, system state, weather, prices, and other contextual information that influence the flexible consumption or generation of electricity (E.L.E.X.O.N., 2021b). The quality, granularity, and timeliness of these data inputs are fundamental to ensuring efficient market functioning and fair settlement of flexibility.

The UK's Smart Metering Equipment Technical Specifications 2 (SMETS2) meters are fundamental instruments for collecting flexibility data (Taskforce, 2019). These advanced meters are capable of two-way communication, allowing them to send detailed energy consumption data to the energy supplier, as well as receive

instructions, such as demand response signals. They record electricity usage at half-hour intervals, which provides the granularity of data necessary for most flexibility services. The SMETS2 meters are interoperable, meaning that they work seamlessly irrespective of the energy supplier, ensuring a smooth customer experience even when switching suppliers (B.E.I.S., 2018). Moreover, these smart meters are essential for establishing the 'actual' energy profile of a flexible asset, facilitating the precise calculation of delivered flexibility (GB, 2021). In essence, SMETS2 meters form the backbone of the data infrastructure that enables the provision and settlement of flexibility in the UK. Their widespread deployment is a testament to the UK's commitment to achieving a more flexible, efficient, and sustainable energy system. The communication set-up is presented in Figure 4:2. This setup illustrates how the electricity and gas smart meters communicate through a central communications hub, which connects to the in-home display and other smart devices over the home area network. The communications hub also links to the energy supplier and DNO through a wide area network, enabling two-way data exchange required for flexibility services. This arrangement ensures real-time visibility and control over household energy usage and is a key part of the digital infrastructure supporting a flexible and efficient energy system in the UK.

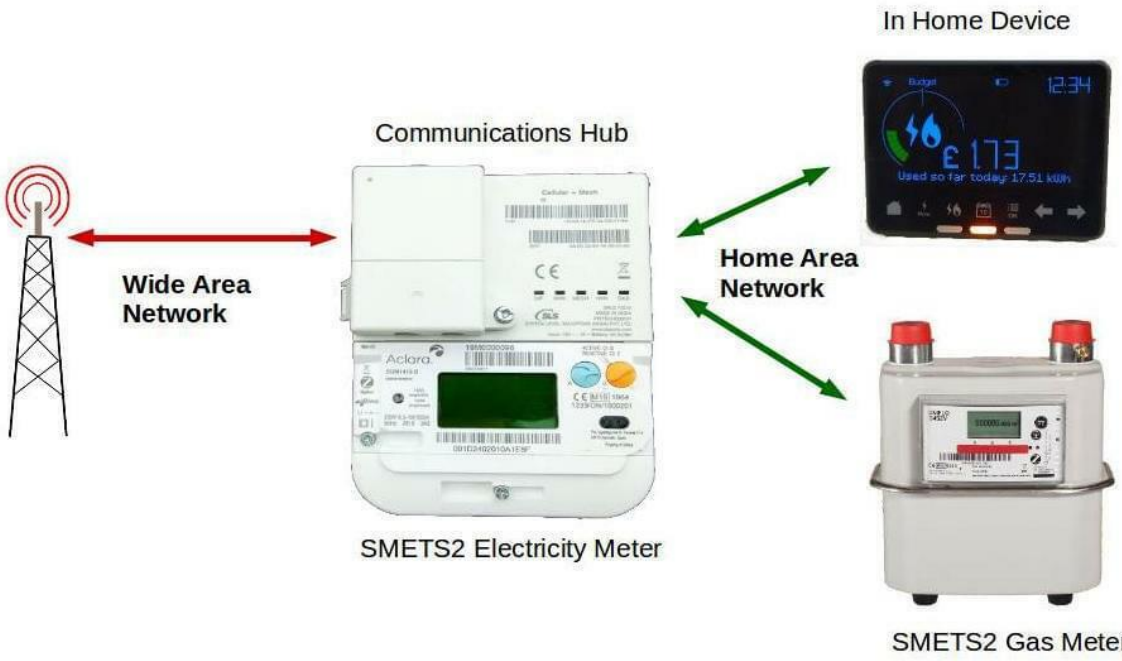


Figure 4:2 Communication network set-up

The baseline energy profile represents a crucial component in the settlement of flexibility services. It provides a 'counterfactual' scenario, essentially showing what the energy consumption or generation would have been under normal, non-flexible conditions. To calculate the baseline, historical energy usage or generation data is required, typically collected from smart meters or other data acquisition systems. This historical data should be granular and high-quality, ideally with readings taken at regular intervals throughout the day.

Once the historical data is collected, different methods can be employed to create the baseline, ranging from simple average calculations to more complex regression models. The chosen method should account for factors such as time of day, day of the week, and seasonal variations, which are known to significantly influence energy patterns. Weather normalisation might also be required for certain types of flexible assets.

The delivered flexibility is calculated by comparing the baseline energy profile with the actual energy profile during the flexibility event (EnerNOC, 2011). The actual energy profile is obtained through real-time or near-real-time monitoring of the asset's energy consumption or generation. By subtracting the actual energy profile from the baseline, one can determine the amount of energy either saved or over-generated due to the flexibility event.

This calculation requires highly accurate and granular data to ensure that the resulting flexibility measurement is precise. It's important to note that this process requires robust data handling procedures to account for potential discrepancies or outliers in the actual energy data. Furthermore, the reconciliation and verification processes necessitate a robust system capable of handling and processing large amounts of data accurately and efficiently. An illustrative example is presented in Figure 4:3, where the area between the baseline (expected consumption) and the actual meter reading represents the delivered flexibility service during a designated event window.

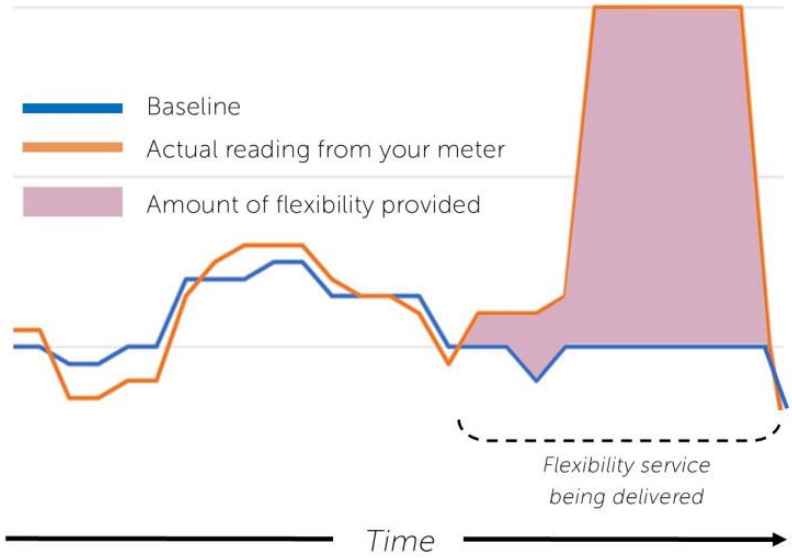


Figure 4:3 Flexibility baseline calculation

In conclusion, while the amount of flexibility delivered and the timeframe are indeed vital pieces of data, the process involves an array of other data requirements. These other factors contribute to the creation of an accurate and fair system for the settlement of flexibility in the UK's energy market.

4.3 A ZKP approach towards flexibility delivery settlement process

Traditional methods of flexibility delivery settlement can benefit from innovative cryptographic methods like ZKP. ZKP is a protocol in which one party (the prover) can prove to another party (the verifier) that they know a specific piece of information, without revealing any details about the information itself. In the context of flexibility delivery settlement, a ZKP approach can offer enhanced security, privacy, and efficiency. It can ensure that settlements are based on verified data, without requiring disclosure of sensitive information, such as detailed energy consumption patterns or the specific operating conditions of flexible assets. This section will delve into how a ZKP approach can revolutionise the settlement process, facilitating a secure, transparent, and efficient system for all parties involved in flexibility services. In this context transparent means with high levels of visibility over the process.

From a cybersecurity perspective, ZKPs offer robust protections. By ensuring that critical or sensitive information is never directly exposed during the verification process, the risk of data breaches or misuse of information is substantially reduced. This represents a considerable benefit to regulators, who are tasked with safeguarding the integrity of the energy market, as well as FPs who are often concerned about the security of their data.

Furthermore, ZKPs can ease the burden of dealing with customer data for DNOs. The zero-knowledge aspect means that DNOs can validate transactions and fulfil their obligations without needing to handle and process raw customer data, reducing their data management workload and minimising the risk of data mishandling. Consequently, ZKPs contribute to a more efficient, secure, and transparent system, benefiting all stakeholders in the flexibility market.

4.3.1 Actor Interaction

In the flexibility delivery settlement process, several key players play distinct roles.

- DNOs: DNOs are responsible for maintaining and operating the distribution network, which delivers electricity from the transmission network to homes and businesses. In the context of flexibility services, DNOs often act as procurers, buying flexibility services to help manage the network efficiently and securely. They are also responsible for validating and settling flexibility transactions. With the transition to DSOs, their role is becoming increasingly active, managing local grid constraints and balancing demand and supply at a more granular level.
- Ofgem: The regulator oversees the entire electricity market, ensuring that it operates efficiently, fairly, and transparently. In terms of flexibility service, the regulator's role includes setting rules and standards for flexibility services, protecting consumers' interests, and ensuring the security and integrity of the market. They also monitor the actions of DNOs and other market players to prevent anti-competitive behaviour and promote a healthy market environment.
- FPs: FPs can be various entities - from large power plants to small residential prosumers with solar panels and batteries, and aggregators who pool smaller resources to provide flexibility services. Their role is to adjust their electricity production or consumption in response to signals from the DNOs or the wider

electricity market. They provide valuable services that help balance the grid, manage network constraints, and enable the integration of more renewable energy. FPs enter into contracts to provide these services, and the fulfilment of their commitments is verified and settled through the flexibility delivery settlement process, as presented in Figure 4:4.

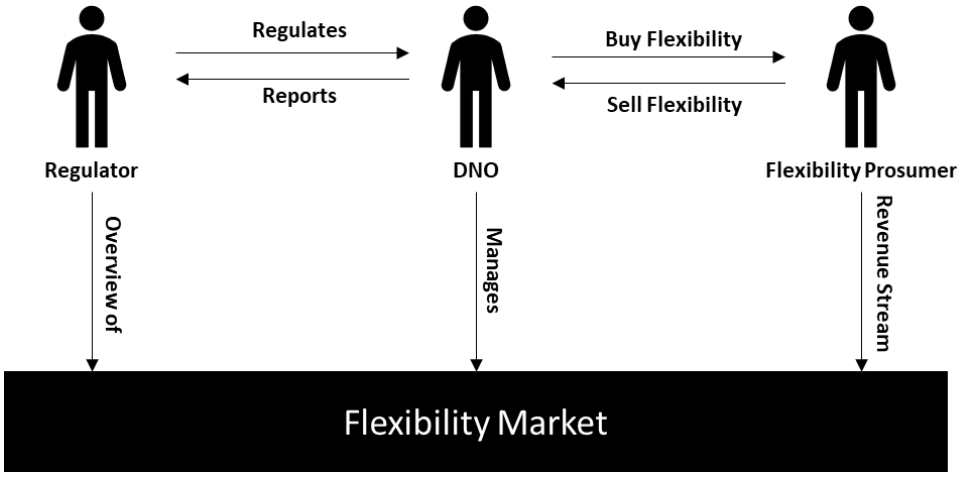


Figure 4:4 Flexibility delivery settlement

4.3.2 Implementation of the Zero Knowledge Proof

As mentioned before, for the settlement of flexibility, the thesis is considering that the flexibility delivered is expressed as the differentiation between the baseline energy profile and the current energy profile. Mathematically, this has been expressed as

$$F(t) = B(t) - A(t), \tag{1}$$

and

$$F_{total} = \sum_{i} F(t)0 = \sum_{i} (B(t) - A(t)) \tag{2}$$

where:

- $F(t)$ represents the delivered flexibility at time t ,
- $B(t)$ is the baseline energy profile at time t ,
- $A(t)$ is the actual energy profile at time t , and
- F_{total} is the total flexibility delivered.

The time period is divided into discrete 15 minutes intervals. For each of these time intervals t, the algorithm calculates the flexibility $F(t)$ by subtracting the actual energy consumption $A(t)$ from the baseline energy consumption $B(t)$. The baseline $B(t)$ is usually through historical data. The actual consumption $A(t)$ is measured in real-time using the flexibility proveder smart meter. The total flexibility $F(t)$ is then computed by summing these differences across all time intervals. This summation provides a comprehensive measure of the overall energy flexibility delivered during the entire period T. It's worth noting that $F(t)$ can be positive (indicating a reduction in consumption) or negative (indicating an increase), allowing for both up and down regulation of energy consumption.

The Pedersen Commitment Scheme is a cryptographic primitive that allows a party to commit to a chosen value while keeping it hidden from others, with the ability to reveal the committed value later. In the context of zero-knowledge proofs for energy flexibility, it's used to create commitments to the baseline and actual energy profiles without revealing the actual values. The scheme operates in a cyclic group g of prime order q , where the discrete logarithm problem is assumed to be hard. Two generators of this group, g and h , are chosen such that the discrete logarithm of h with respect to g is unknown to anyone. For each time interval t , commitments are created to the baseline $B(t)$ and actual $A(t)$ energy values:

$$C_{B(t)} = g^{B(t)} \times h^{r_{B(t)}} \, mod \, p \tag{3}$$

$$C_{A(t)} = g^{A(t)} \times h^{r_{A(t)}} \, mod \, p \tag{4}$$

Here, $r_{B(t)}$ and $r_{A(t)}$ are random values, inheriting from the set of integers $mod \, p$ where q is a random prime number and allows maintaining the security properties of the commitment scheme, for each commitment. These random values serve as "blinding factors" that hide the committed values. The modular arithmetic is performed with respect to a large prime $\boldsymbol{p}$, which defines the order of the group $\boldsymbol{G}$. This ensures that the commitments are elements of the group. The Pedersen Commitment Scheme has two properties:

- Hiding: The commitment C does not reveal any information about the committed value. This is due to the random factor r , which masks the committed value.
- Binding: It is computationally infeasible for the committer to find two different pairs (value, random factor) that produce the same commitment. This property relies on the discrete logarithm assumption.

Furthermore, the Pedersen Commitment Scheme is homomorphic, meaning that operations on the committed values can be performed on the commitments themselves. Specifically:

$$C_{A(t)} \times C_{B(t)} = (g^{A(t)} \times h^{r_1}) * (g^{B(t)} \times h^{r_2}) = g^{(A(t)+B(t))} \times h^{r_2 * r_1} = C_{(A(t)+B(t))} \quad (5)$$

This homomorphic property allows the performance of calculations on the committed values without revealing them.

In the context of this chapter, polynomial representation is a technique used to encode the baseline, actual, and flexibility energy profiles. This approach allows to leverage the properties of polynomials to create efficient and succinct proofs. Now, let's represent $A(t)$, $B(t)$, and $F(t)$ as polynomials:

$$B(x) = \sum_{i}^{0} B(t_i) \times L_i(x) \quad \textbf{(6)}$$

$$A(x) = \sum_{i}^{0} A(t_i) \times L_i(x) \quad \textbf{(7)}$$

$$F(x) = B(x) - A(x) \quad \textbf{(8)}$$

Here, $L_i(x)$are Lagrange basis polynomials. The Lagrange basis polynomials are a set of polynomials that have a special property: $L_i(t_j)$= 1 if i = j, and 0 otherwise. This property allows to interpolate a polynomial that passes through all the points (t_i, $B(xt_i)$) or (t_j, $B(xt_j)$) . The Lagrange basis polynomials are defined as:

$$L_i(x) = \prod_{j!=i} \frac{x - t_j}{t_i - t_j} \quad \textbf{(9)}$$

To express the computation as a Quadratic Arithmetic Program (QAP), formulate the statement: "I know polynomials $A(x)$ and $B(x)$ such that":

$$C_B = \prod_t (g^{B(t)} \times h^{r_B(t)}) \quad \textbf{(10)}$$

$$C_A = \prod_t (g^{A(t)} \times h^{r_A(t)}) \quad \textbf{(11)}$$

The QAP structure involves polynomials $v_k(x)$, $w_k(x)$, $y_k(x)$ for k∈{0…,m}, where m is the number of gates in the arithmetic circuit. The QAP is satisfied if:

$$\left(\sum_k a_k v_k(x)\right) \times \left(\sum_k a_k w_k(x)\right) - \left(\sum_k a_k y_k(x)\right) = h(x)t(x) \quad \textbf{(12)}$$

Where:

- a_k are the wire values in the arithmetic circuit.
- $h(x)$ is the quotient polynomial.
- $t(x)$ is the target polynomial $(x-1)(x-2)\cdots(x-n)$.

Let $e : G_1 \, x \, G_2 \rightarrow G_t$ be a bilinear map, where:

- $G_1$and $G_2$are additive groups of prime order r, typically represented by points on an elliptic curve.
- G_t is a multiplicative group of the same order r, typically represented by elements of a finite field extension.

The bilinear pairing e has the following key properties:

- Bilinearity: For all a, b ∈ Z_r and $P \in G_1, Q \in G_2$: $e(aP, bQ) = e(P, Q)^{ab}$
- Non-degeneracy: If $e(P, Q)$= 1 for all $Q \in G_2$, then P = 0. Similarly, if $e(P, Q)$ = 1 for all $P \in G_1$, then ç = 0.
- Efficiency: The pairing e can be computed efficiently.

These properties allow to "move" exponents between the groups and perform certain equality checks efficiently, which is crucial for the succinctness and zero-knowledge properties of the proof system.

The setup phase is a critical component of zk-SNARK systems. It generates the public parameters that will be used for creating and verifying proofs. This phase is often referred to as the "trusted setup" because it requires a trusted party or a secure multi-party computation protocol to generate these parameters. The trusted setup generates two key components:

- Proving Key (p_k)
- Verification Key (v_k)

The proving key pk consists of the following elements:

- $(\alpha, \beta, \gamma, \delta)$: Random elements in F_r (the field of order r)
- $\{x^i\}_{i=0}^{n}$: Powers of a random element $x \in F_r$
- $\{\frac{\beta \mathrm{v}_k(x)+\alpha \mathrm{w}_k(x)+\mathrm{y}_k(x)}{\gamma}\}_{k-0}^{m}$: Combination of QAP polynomials
- $\{\frac{x^i}{\delta}\}_{i=0}^{n}$: Powers of x divided by δ

The verification key vk consists of:

- $(\alpha, \beta, \gamma, \delta)$: The same random elements as in the proving key
- $\{g^{x^i}\}_{i=0}^{n}$: Generator g raised to powers of x

The randomness used in this phase $(\alpha, \beta, \gamma, \delta)$ must be discarded after setup. If an adversary learns these values, they could create false proofs. The setup phase is circuit specific. Different circuits (i.e., different statements to be proven) require different setups. The security of the entire zk-SNARK system relies on the integrity of this setup phase. The proving key pk is used by the prover to generate zero-knowledge proofs. The verification key vk is used by the verifier to check the validity of proofs. The structure of these keys allows for efficient proof generation and verification while maintaining the zero-knowledge property.

In the proof generation phase, the prover uses the proving key pk to create a succinct proof that they know a valid assignment to the QAP (i.e., valid $B(x)$ and $A(x)$ polynomials) without revealing these polynomials. The prover computes three main components:

$$A = g_1^{(\sum_k a_k v_k(x))} \tag{13}$$

$$B = g_2^{(\sum_k a_k w_k(x))} \tag{14}$$

$$C = g_1^{(\sum_k a_k y_k(x) + h(x)t(x))/\delta} \tag{15}$$

Where:

- g_1 and g_2 are generators of $G_1$and G_2 respectively
- a_k are the wire values in the arithmetic circuit
- $v_k(x), w_k(x), y_k(x)$ are the QAP polynomials
- $h(x)$ is the quotient polynomial
- $t(x)$ is the target polynomial
- δ is a random value from the setup phase

The zk-SNARK proof π is then composed of these three elements: $\pi = (A, B, C)$. This proof is succinct (constant-sized regardless of the complexity of the statement being proved) and zero-knowledge (it reveals nothing about the actual values of $B(x)$ and $A(x)$).

The verification process allows a verifier to check the validity of the proof π without learning anything about the prover's secret information ($B(x)$ and $A(x)$ in this case). The verifier checks the following equation using the verification key vk and the bilinear pairing e:

$$e(A, B) = e(\alpha, \beta) \times e(C, g_2^{\alpha}) \times e\left(\prod_i g_1^{v_i(x)p_i}, g_2\right) \times e(g_1, \prod_i g_2^{w_i(x)p_i}) \tag{16}$$

Where:

- p_i are the public inputs (including the commitments $C(B)$ and $C(A)$, and F_{total}
- $\alpha, \beta, \gamma, \delta$ are from the verification key
- g_1 and g_2 are generators of G_1 and G_2 respectively

If this equation holds, the verifier accepts the proof as valid. This check ensures that the prover knows a valid assignment to the QAP that is consistent with the public inputs, without revealing any information about the secret inputs.

Recursive composition is a powerful technique used in the MINA protocol to achieve constant-sized proofs regardless of the computation size. It allows for the creation of proofs that verify the correctness of other proofs, forming a chain of verifications.

The fundamental concept is that:

$$\pi_{i+1} = prove\ (pk, \{\pi_i, x_{i+1}\}, w_{i+1}) \quad \textbf{(17)}$$

Where:

- π_i is the proof of the correctness of the computation up to step i
- x_{i+1} is the public input for step i +1
- w_{i+1} is the witness (secret input) for step i +1

This recursive structure allows the ZKP blockchain to compress arbitrarily large computations into a fixed-size proof, making it particularly suitable for blockchain applications where space efficiency is important.

The homomorphic properties of Pedersen commitments allow for efficient aggregation of commitments, which is crucial for the energy flexibility verification system. For the baseline and actual energy profiles, the next step is to aggregate the commitments:

$$C_B^{total} = \prod_t C_B(t) = g^{\sum_t B(t)} \times h^{\sum_t r_B(t)} \quad \textbf{(18)}$$

$$C_A^{total} = \prod_t C_A(t) = g^{\sum_t A(t)} \times h^{\sum_t r_A(t)} \quad \textbf{(19)}$$

This allows the DSO to verify the total flexibility without learning individual values:

$$F_{total} = log_g(\frac{C_B^{total}}{C_A^{total}}) \quad \textbf{(20)}$$

In practice, computing this discrete logarithm efficiently can be done using algorithms like the baby-step giant-step algorithm or Pollard's rho algorithm. This aggregation property is crucial for scalability, as it allows the system to handle large numbers of time intervals efficiently.

The security of this zero-knowledge proof system for energy flexibility relies on several key properties:

- Completeness: An honest prover who knows valid $B(x)$ and $A(x)$ polynomials can always generate a proof that will convince the verifier.
- Soundness: It is computationally infeasible for a dishonest prover to generate a valid proof for an invalid statement. This property relies on the hardness of the discrete logarithm problem in the groups used.
- Zero-knowledge: The verifier learns nothing about the prover's secret information ($B(x)$ and $A(x)$) beyond what is implied by the validity of the statement being proved.

The security of the system also relies on cryptographic assumptions such as the hardness of the discrete logarithm problem and the bilinear Diffie-Hellman assumption in the groups used.

4.3.3 System Architecture

The implementation of this solution hinges upon leveraging the MINA blockchain as the fundamental platform. Renowned for its use of recursive zk-SNARKs, MINA blockchain empowers efficient verification of transactions without necessitating a full node. This translates to network participants being able to verify transactions without the storage of the complete blockchain, thereby providing significant scalability and efficiency advantages.

In the context of this solution, FPs construct a proof on the MINA blockchain demonstrating the delivery of the required energy amount, all while keeping their energy profile details concealed. This proof can then be forwarded to the network operator who, using the MINA blockchain, can validate the proof's legitimacy, thereby preserving the privacy and security of the provider's data without needing to access their energy profile, as presented in Figure 4:5.

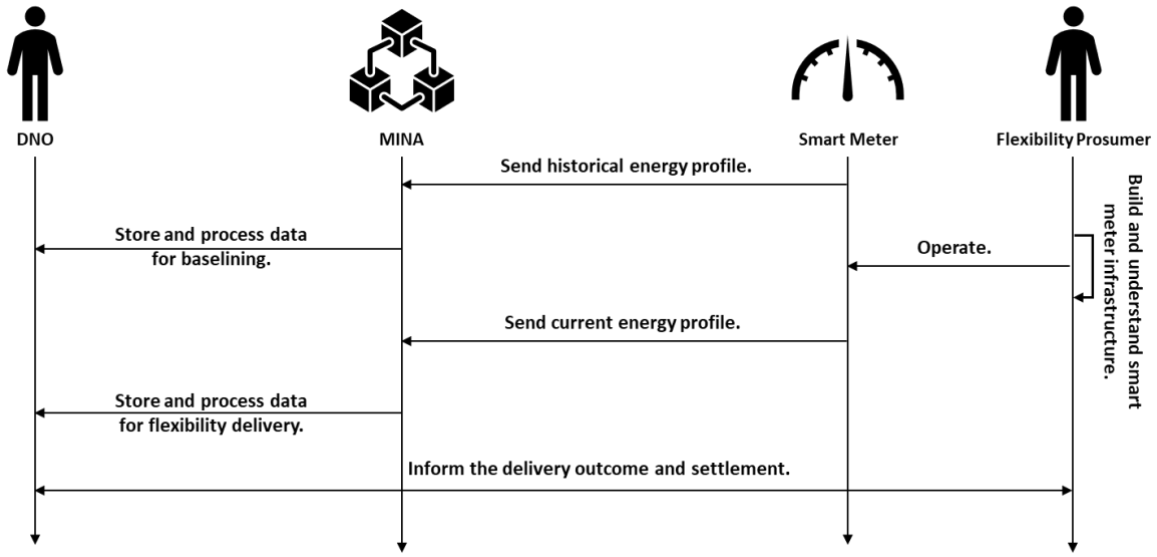


Figure 4:5 System architecture sequence diagram

To maintain the anonymity of the asset providing the flexibility - a measure aimed at preventing premeditated attacks to obstruct delivery - on-chain mapping between the geographical location and the identification of the asset will remain confidential. Off-chain communication will occur between the DNO and the winning bidder of the flexibility service, with the flexibility contract's smart contract securely containing an immutable hash of the winner's private identification key. After completion confirmation, the prosumer or provider will have the capability to withdraw the financial settlement, remaining anonymous to others viewing the on-chain data.

To prove the technology is faster than the future requirements of half hour settlement, proofs will be created every 15 minutes. Considering the overarching energy sector's objective of achieving 15-minute data granularity, the thesis does not anticipate tighter data granularity requirements in the foreseeable future.

The implementation of ZKPs in the MINA blockchain offers distinct advantages over the current flexibility settlement procedure. Primarily, it boosts the privacy and security of the FP, ensuring the confidentiality of their energy profile. It also simplifies the settlement process for both the FP and the network operator, lessening the administrative workload and reducing transaction expenses. Moreover, it paves the way for the expansion of the flexibility market, bolstering the security and privacy of participants and accelerating the shift towards a more sustainable, decentralised energy system, as presented in Figure 4:6.

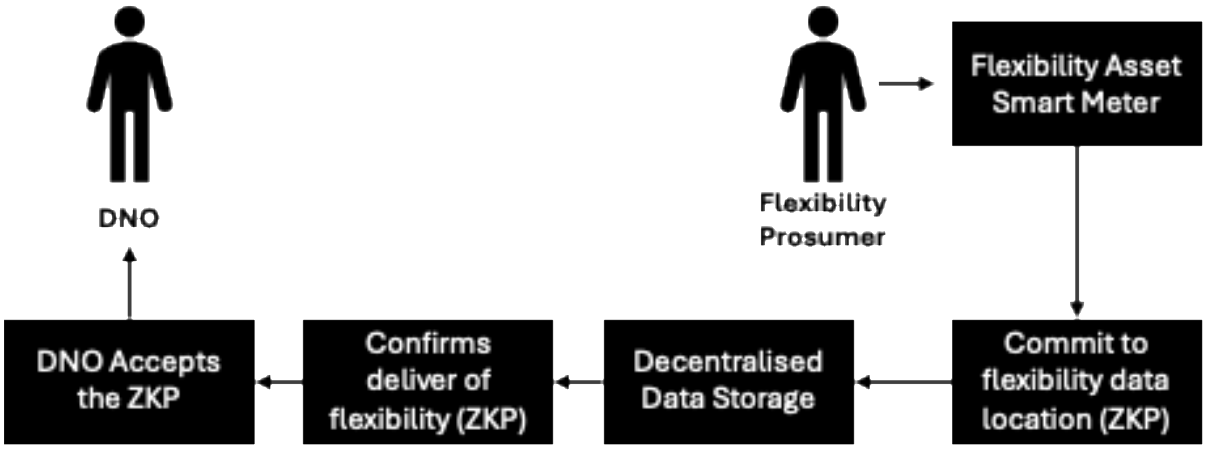


Figure 4:6 Implementation of ZKP

The proposed zkApp design and architecture revolve around a smart contract that facilitates the settlement process. The prosumer's client (which is the hardware technology pack that sits, and it is owned by the prosumer) generates a proof of flexibility delivery, without disclosing sensitive information regarding the prosumer's energy profile, in accordance with the constraints of the smart contract. Consequently, the network operator can validate this proof and finalise the contract with the prosumer.

The interaction between these components proceeds as follows:

- The prosumer's energy assets and metering devices generate and record data on energy consumption and production.

- The prosumer initiates a flexibility transaction, specifying their offered flexibility, with the network operator's flexibility management system.
- The network operator's flexibility management system validates the transaction and records it on the MINA blockchain.
- The MINA blockchain utilises zk-SNARKs to validate the transaction without exposing any data about the prosumer's energy profile.
- The network operator compensates the prosumer for the delivered flexibility, based on the verified transaction recorded on the MINA blockchain.

When the DNO) needs to settle flexibility delivery, zero-knowledge (ZK) proofs will be employed to generate two types of proofs.
The first proof confirms that the correct data has been uploaded to the decentralised storage, ensuring that the data recording process has been followed accurately and that the recorded data is consistent with the agreed-upon parameters. The second proof verifies that the difference between the baseline and the actual energy profile equals the amount of flexibility that was delivered. This proof confirms that the prosumer has delivered the correct amount of flexibility in accordance with their commitments.

These proofs will then be submitted to the DNO. The DNO can then check the validity of the proofs using the unique properties of ZK proofs, which allows them to verify the correctness of the information without needing to know the information itself. If both proofs are valid, the DNO can confidently settle the contract and pay the prosumer for the flexibility they provided. However, if the prosumer hasn't delivered the agreed-upon flexibility, the DNO can impose penalties as per the terms of the contract. This system ensures automated accountability in managing energy flexibility in the network, as presented in Figure 4:7.

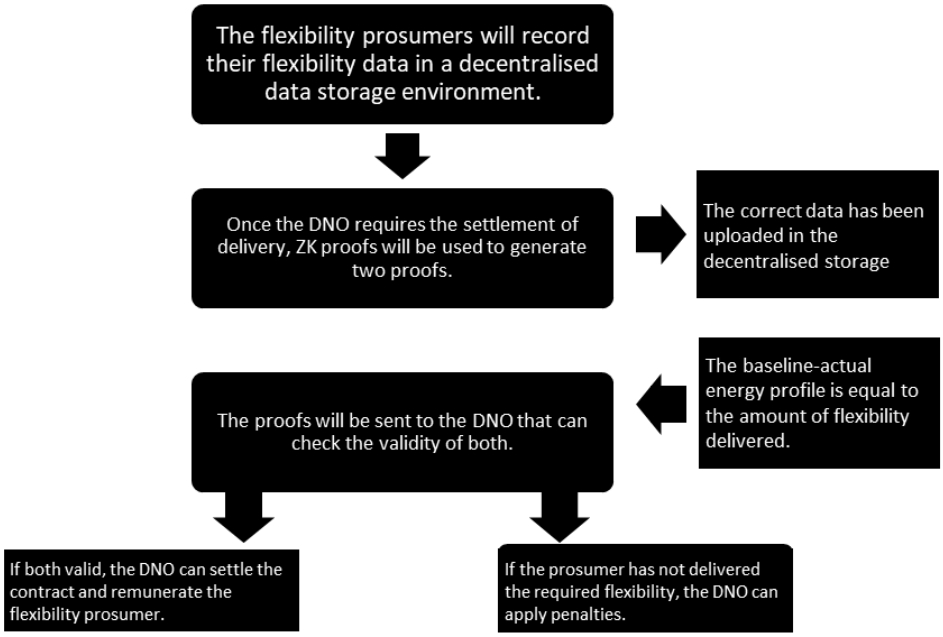


Figure 4:7 Flexibility ZKP settlement sequence

4.3.4 Implementation

The implemented solution for flexibility services in the energy market, smart contracts play a vital role in streamlining and securing transactions between prosumers and network operators. These contracts are built on the MINA blockchain, written using snarky.js.

The "Registration of Energy Assets" smart contract serves as the entry point for prosumers into the flexibility ecosystem. It enables them to register various types of energy assets—such as solar panels, EVs, and battery storage systems—with the network operator's system. The smart contract will store essential details like the type of asset, its capacity, location, and unique identifier. The logic encapsulated within this contract ensures that each asset is distinctively identified, and its capabilities are accurately logged for future interactions. This process forms the basis for matching prosumer assets with the flexibility needs of the network operator, setting the stage for the negotiation and formulation of flexibility contracts.
The functionality description is presented in Table 4:2.

Key Element	Description
Smart Contract Function	`registerAsset(assetDetails)`
Description	Allows the prosumer to register their energy asset (e.g., solar panel, electric vehicle, etc.) in the system.
Parameters	`assetDetails` containing the type, location, capacity, etc., of the energy asset.
Privacy Measures	Utilizes zk-SNARKs to maintain the anonymity of the prosumer while verifying the legitimacy of the asset.

Table 4:2 Registration of Energy Assets

Once an energy asset is registered, the "Flexibility Contract Creation" smart contract comes into play. This contract manages the negotiation phase between the prosumers and the network operators. Prosumers can submit their offers, specifying parameters like the amount of flexibility they can provide, the time frames, and the rate at which they wish to be compensated. The network operator can review these offers and either accept, counter, or reject them. Once both parties reach an agreement, the smart contract will solidify the flexibility contract, often represented as a digital agreement bound by specific terms and conditions. This smart contract directly interfaces with the Registration smart contract to ensure only registered assets can engage in flexibility contracts. The functionality description is presented in Table 4:3.

Key Element	Description
Smart Contract Function	`createFlexibilityContract(contractDetails)`
Description	Enables the negotiation and creation of flexibility contracts between the prosumer and the network operator.
Parameters	`contractDetails` encompassing the amount of flexibility, duration, price, etc.
On-chain Mapping & Secrecy	An immutable hash of the private key of the winning bidder is set to protect the identity of the asset.

Table 4:3 Flexibility Contract Creation

After the flexibility contract is in place, the "Proof of Flexibility Delivery" smart contract becomes crucial. This contract uses zk-SNARKs to validate that the prosumer has indeed fulfilled their part of the contract by delivering the agreed-upon flexibility. The prosumer's client generates a proof that they have complied with the agreed parameters, and this proof is then verified by the smart contract. Importantly, the proof is constructed such that it verifies the contract's fulfilment without revealing sensitive information about the prosumer's energy usage or production patterns. This maintains the privacy and confidentiality that are integral to the system. The functionality description is presented in Table 4:4.

Key Element	Description
Smart Contract Function	`generateFlexibilityProof(transactionDetails)`
Description	Validates the delivery of flexibility by creating a proof without revealing sensitive information.
Parameters	`transactionDetails` containing information about the delivery of the contracted flexibility.
Verification Process	The proof is submitted to the network operator, who verifies its validity using the MINA blockchain.

Table 4:4 Proof of Flexibility Delivery

The final stage of the process is governed by the "Settlement and Compensation" smart contract. This contract is activated once the Proof of Flexibility Delivery smart contract confirms that the prosumer has met the contract's requirements. It automates the transfer of funds or credits from the network operator to the prosumer, thereby ensuring timely and accurate compensation for the flexibility services provided. The logic here also checks for any conditions or penalties that might apply based on the performance metrics set in the flexibility contract. The functionality description is presented in Table 4:5.

Key Element	Description
Smart Contract Function	`settleContract(contractID, compensationAmount)`
Description	Manages the settlement and compensation for the delivered flexibility.
Parameters	`contractID` identifying the specific contract, and `compensationAmount` detailing the payment due.
Transaction Security	Incorporates zk-SNARKs for secure and efficient payment without exposing detailed transaction data.

Table 4:5 Settlement and Compensation

These smart contracts don't operate in isolation; they are intrinsically. The Registration contract feeds into the Flexibility Contract Creation contract, which in turn connects to the Proof of Flexibility Delivery contract. Once proof is verified, the Settlement and Compensation contract is invoked to complete the transaction loop.

4.4 Test case and results

4.4.1 Network description

This LV network comprises six FPs, each operating two types of DERs: a PV system and a Battery Storage System, represented here by an EV. The network design is based on National Grid Electricity Distribution's Standard Technique SD5A/6 (Pope and Treasure, 2021) which outlines the requirements for the design of LV domestic connections. This document was selected as the foundation for this study due to its alignment with UK regulatory standards and its provision of practical parameters for demand estimation, import capacities, and the integration of low-carbon technologies such as EVs and PVs within a residential setting.

The LV network configuration is illustrated in Figure 4:8. It represents a simplified, residential feeder where multiple DERs are integrated into a coordinated flexibility framework. Each provider contributes to the network's operation through their PV and battery assets, supporting a decentralised approach to energy balancing. This model reflects an emerging grid structure that leverages data and distributed technologies to improve reliability, efficiency, and sustainability. By offering ancillary

services, such as voltage support and peak load reduction, these decentralised assets play a vital role in the stability and flexibility of the grid.

To facilitate simulation and analysis, several simplifications were applied to the network model. Most significantly, the PV generation and battery profiles were replaced with uniform, standardised curves across all providers. This was done to remove variability caused by local weather conditions or system-specific differences, enabling a clearer assessment of flexibility delivery. The simplification allows the study to focus specifically on the performance of flexibility services under consistent solar conditions, while maintaining dynamic interactions for storage.

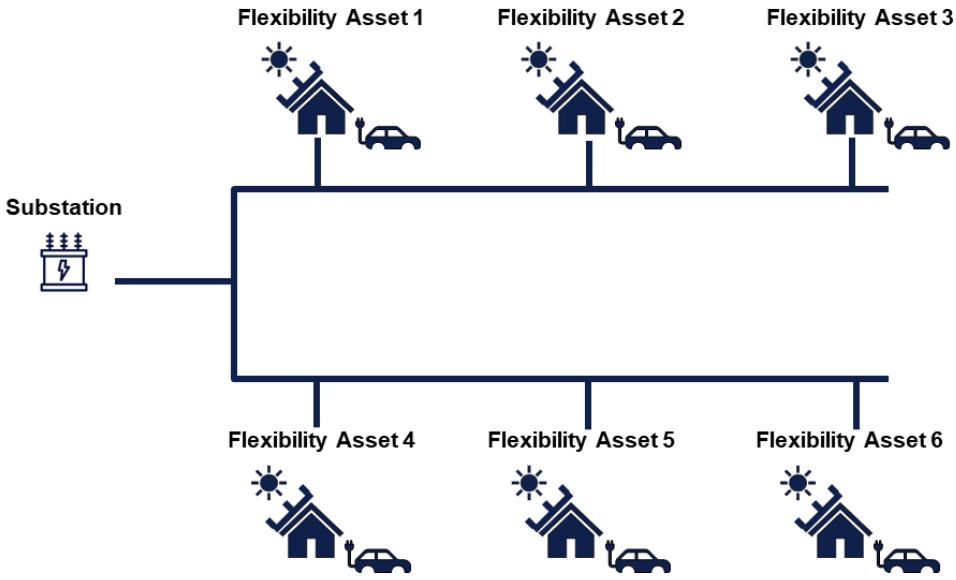


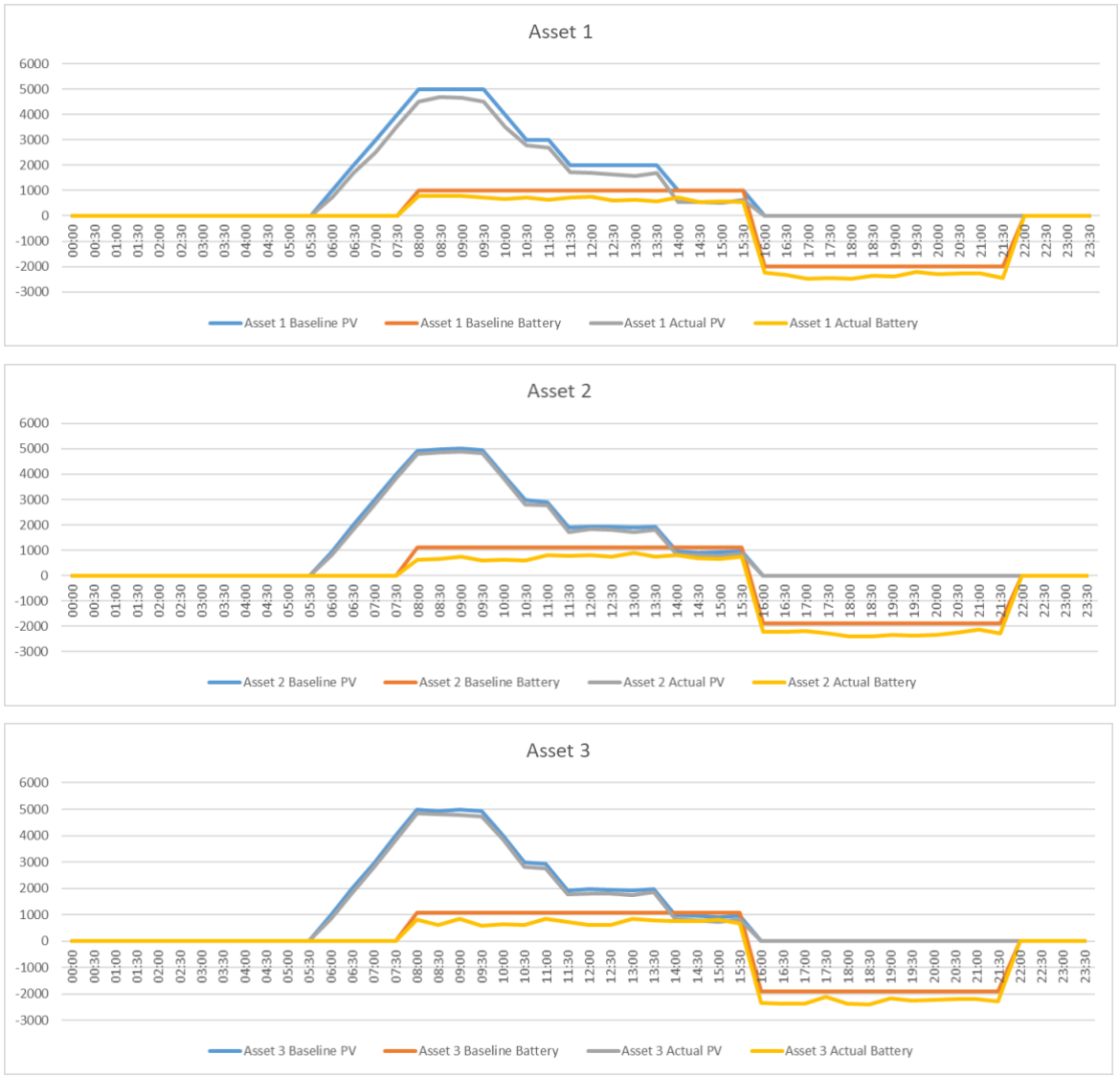
Figure 4:8 LV Network

A typical summer week was selected for the analysis, as this season provides extended daylight hours and higher household energy consumption—conditions which strongly influence PV system output and battery usage. The analysis relies on two energy profiles: a baseline profile and an actual profile for the selected week.

The baseline profile was calculated as the average energy generated and consumed over the preceding three weeks. This approach smooths out any anomalies and provides a robust point of comparison. The actual profile reflects the real-time energy performance of each system during the week in question. Only the PV output follows

a simplified, standardised curve, while the battery system operates based on realistic charging and discharging dynamics.

These profiles are illustrated in Figure 4:9, which presents the data for a representative household for all the assets. The figure compares the baseline and actual PV generation, as well as the associated battery activity. The battery data shows how storage systems respond to generation and demand in real time, with charging occurring during solar peaks and discharging during periods of low generation or higher consumption. The PV and battery data shown are adapted from NESO's Data Portal to represent typical conditions during a summer day.



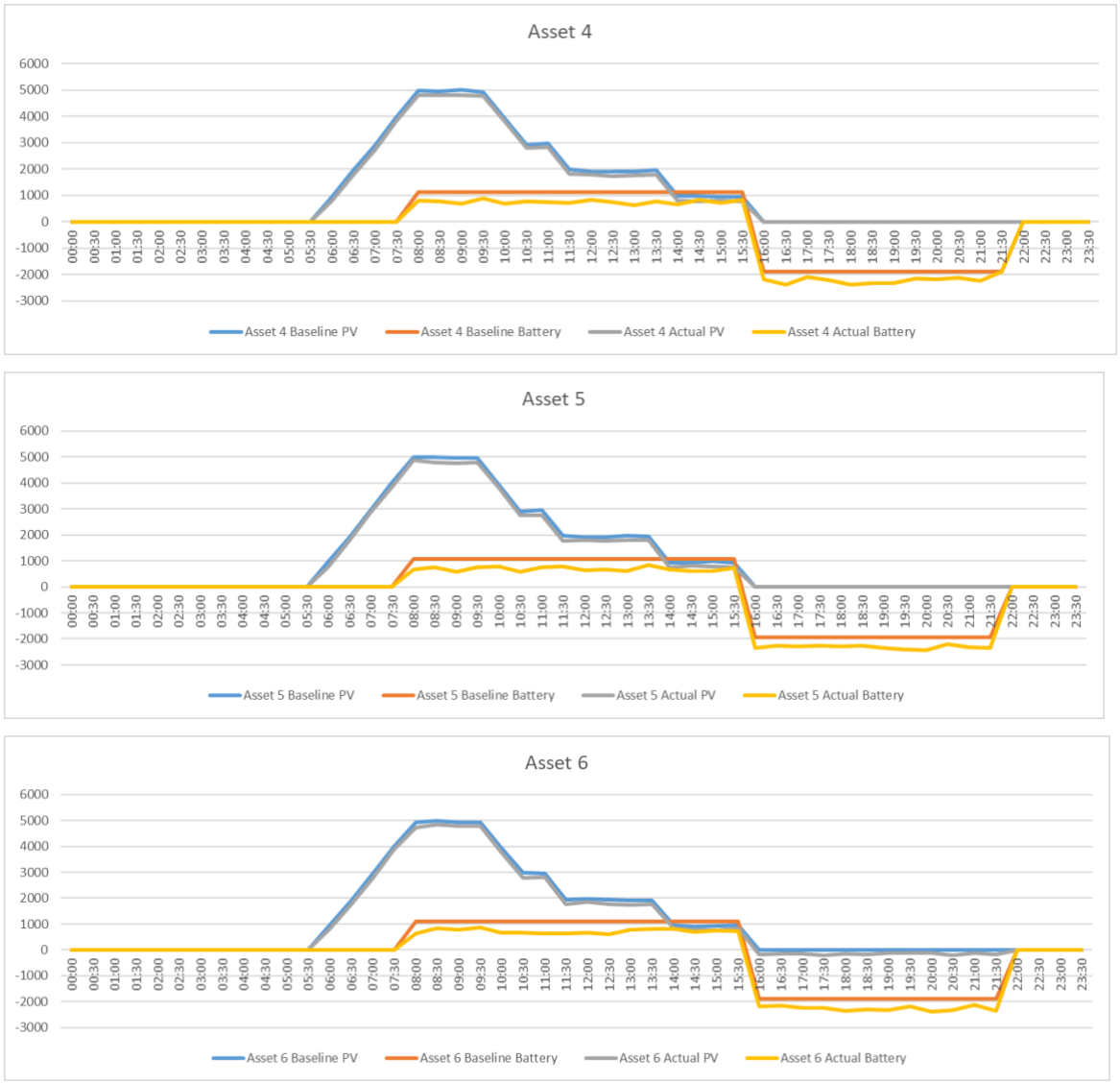


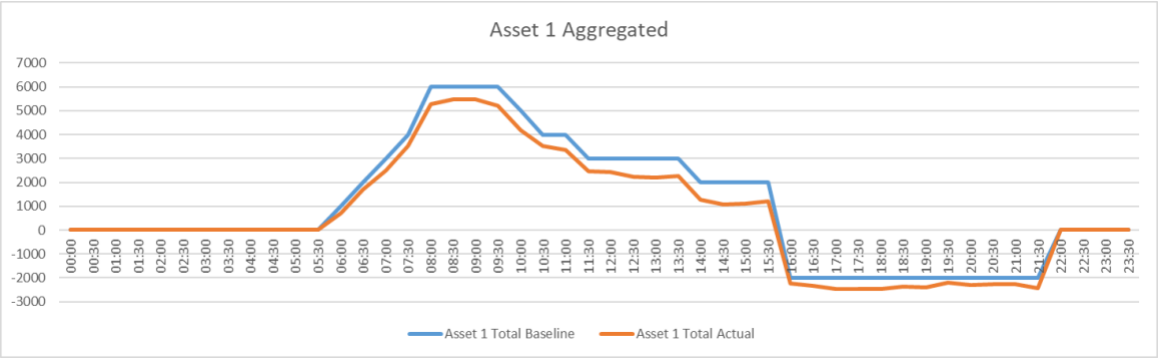
Figure 4:9 All assets baseline and actual profiles (measured in kW)

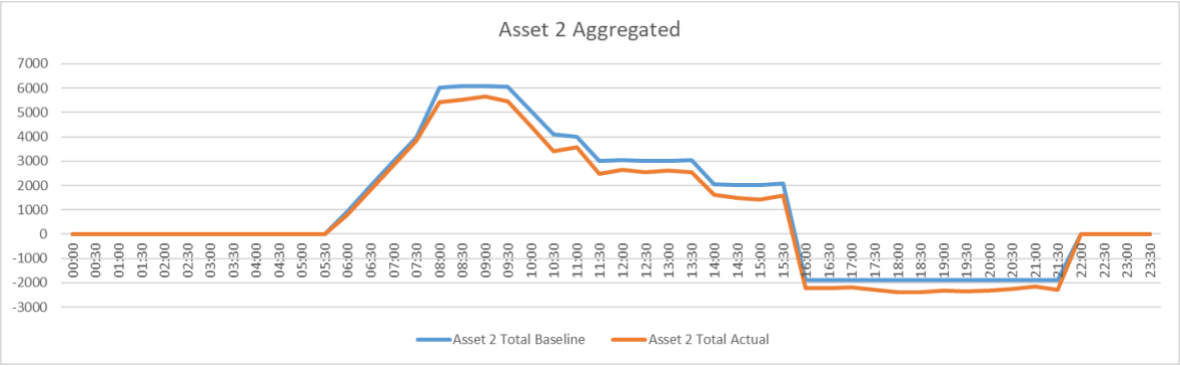
By comparing the actual energy profile with the baseline, the measurement and the performance and reliability of the FPs are calculated. This comparison allows to verify whether the expected flexibility delivery has been met, thereby supporting the settlement processes with the DNO. Following this, insights to further optimise the energy systems and improve their efficiency and reliability have been drafted. The next step is to calculate the aggregated baseline and actual energy profiles for each flexibility prosumer, integrating the photovoltaic system with the battery storage.

In this context, the term "aggregated" refers to the net power consumption of each asset, which accounts for both energy generation and consumption over time. Figure 4:10 presents this data for Asset 1 to 6. During the early hours of the day, both the

baseline and actual profiles remain flat, indicating minimal net consumption or generation. As the day progresses, particularly after 06:30, both profiles begin to rise, reflecting increased energy demand or a scheduled dispatch of flexibility. The baseline peaks at approximately 6,800 watts, while the actual profile slightly underperforms, peaking just below 6,500 watts. This discrepancy may indicate a shortfall in flexibility delivery during the peak period.

Later in the day, between 13:30 and 18:00, the actual profile consistently trails the baseline, possibly suggesting reduced output from generation or storage assets, or a deviation from the expected schedule. From 18:30 onward, both profiles exhibit a sharp drop, transitioning into negative values. This implies net energy export, likely due to excess photovoltaic generation or strategic battery discharge. Notably, the actual profile closely follows the baseline during this period, highlighting good tracking accuracy and a well-executed flexibility response. A final adjustment phase is visible between 21:30 and 23:00, where both profiles return to near-zero levels, indicating stabilisation as the day concludes.





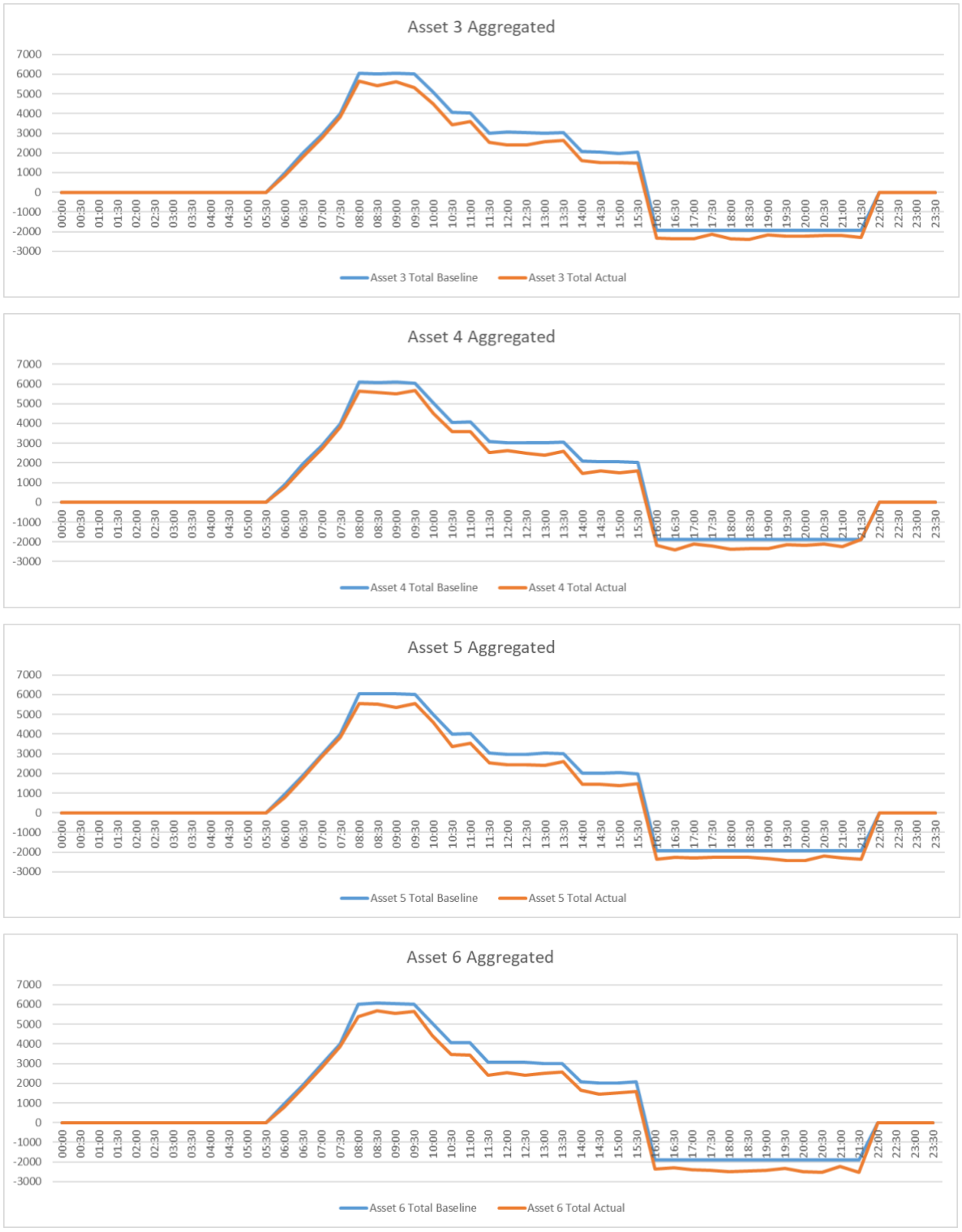


Figure 4:10 All assets aggregated profiles

4.4.2 Analysis

It was assumed the DNO to issue three flexibility contracts for the 24 hours period, each with half hourly requirements of flexibility, as presented in Figure 4:13. The FPs

are assumed to bid for each contract, using a constant price expressed as £/kW, as presented in Figure 4:11.

Figure 4:11 Bid prices

After all data has been imported in the solution, the market clearing algorithm presents FP 1 to be the winner of Contract 1, FP 4 to be the winner of Contract 2 and FP 6 to be the winner of Contract 3, as expected.

Moreover, the following bids have been imported in the platform, for each FP. Figure 4:12 visualises the imported flexibility bid profiles from each of the six DERs acting as FPs. Each line represents the time-varying capacity that an asset has committed to provide in response to the DNO's needs. The figure demonstrates that while the bid patterns generally follow the typical load curve, there are noticeable variations in capacity and timing across assets. This reflects both the operational characteristics of the assets and the autonomous bidding strategies configured in the model. Such granular time-series data is crucial in simulating realistic market conditions and understanding how different assets contribute under various flexibility signals. To contextualise the demand side of the market, Figure 4:13 illustrates the required flexibility over time, as requested by the DNO, disaggregated by contractual obligations. Each line represents a separate flexibility contract, revealing periods of higher demand—particularly during early evening hours, consistent with expected peak load scenarios. The figure highlights the complexity of coordinating multiple contracts and the importance of aligning bids with demand in order to minimise system imbalance. The variability of the demand profile also serves as a useful benchmark for assessing the responsiveness of the aggregated bids and identifying under- or over-procurement scenarios.

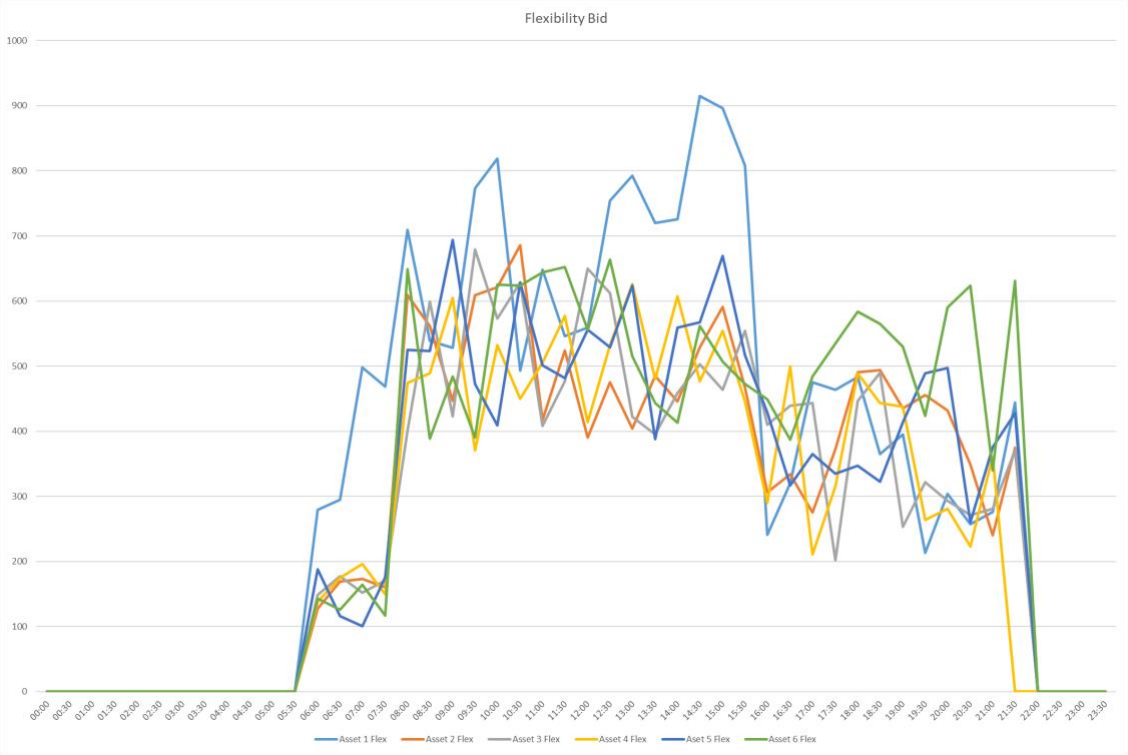


Figure 4:12 Asset bids (expressed in kW)

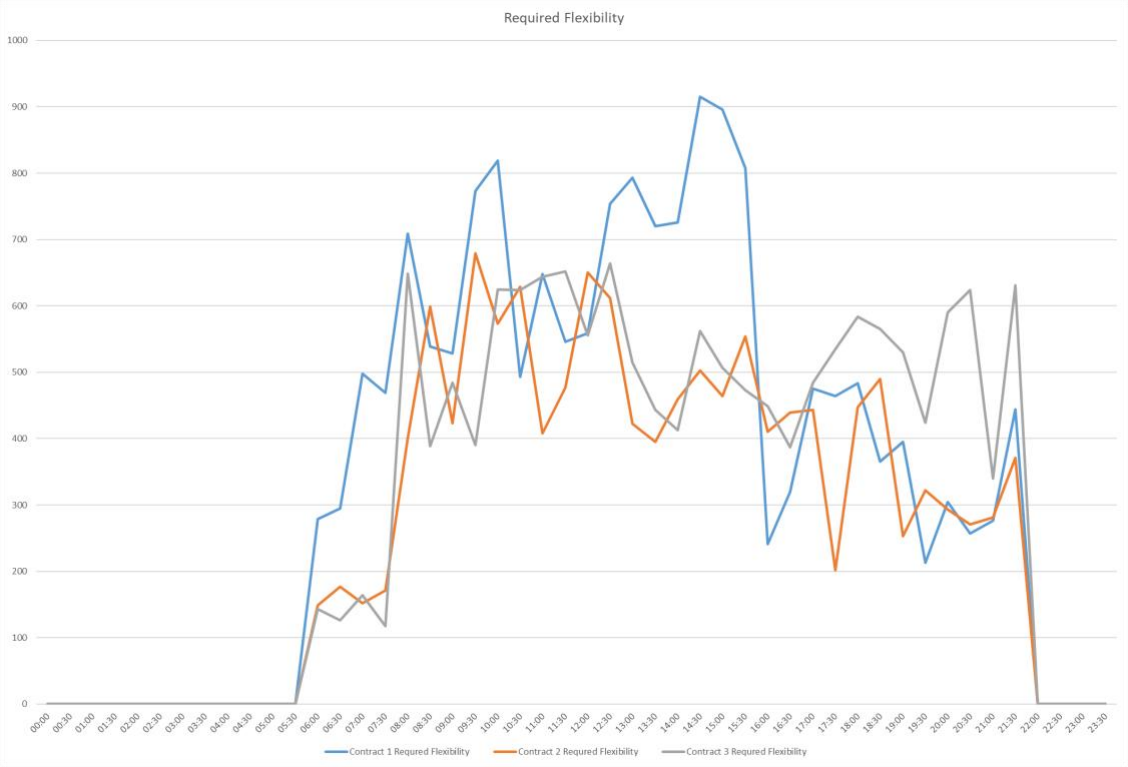


Figure 4:13 Requested flexibility (expressed in kW)

Figure 4:14 offers a comparison between the flexibility that was required and that which was successfully delivered. The shaded areas and line plots show the delivered flexibility from selected assets against the contractual requirements across

time. This comparison provides an evidence-based view of the effectiveness and accuracy of the bidding and dispatch mechanism implemented in the model. While most of the flexibility needs appear to be met, there are time intervals—particularly during sharp peaks—where some shortfalls are evident. This mismatch could stem from forecasting errors, asset constraints, or bid saturation and highlights the importance of improved coordination and incentive alignment in decentralised flexibility platforms. Evaluating this performance supports more accurate settlement procedures and informs the future design of incentive structures.

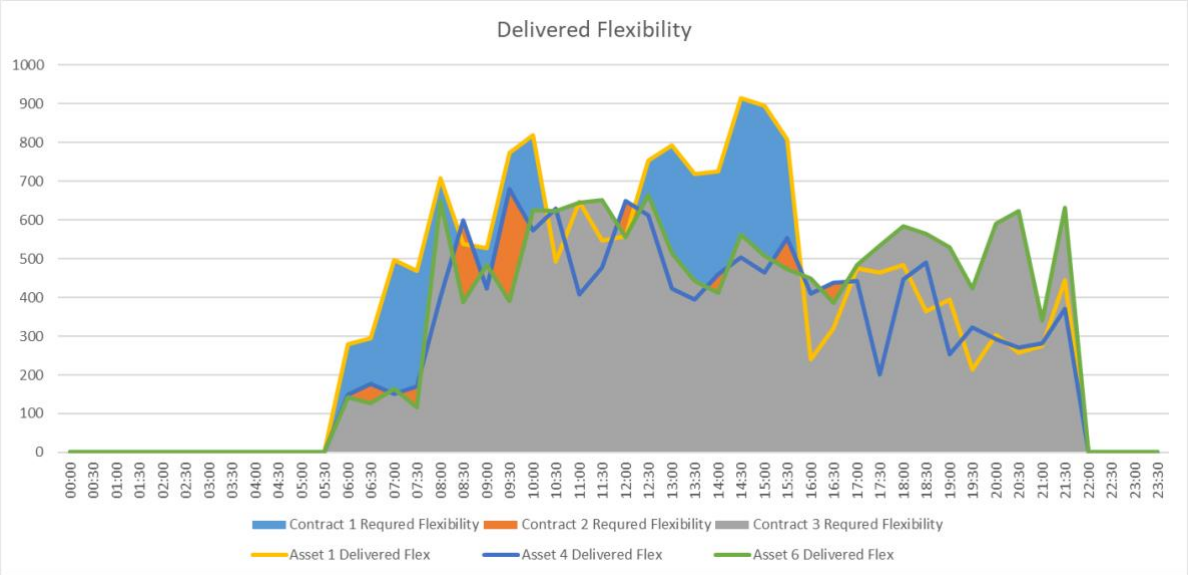


Figure 4:14 Delivered flexibility (expressed in kW)

The results of this chapter demonstrate that a DLT-based framework can support the deployment of a local flexibility market with a high degree of transparency, autonomy, and operational integrity. The prototype implemented on a blockchain platform was able to replicate the full market lifecycle—registration, bidding, dispatch, and settlement—using smart contracts without the need for centralised control. Specifically, the framework enabled automatic validation of flexibility bids and secure tracking of transactions.

Compared to a centralised architecture, the blockchain approach reduced manual handling and the potential for single points of failure. This is a significant advantage for DNOs managing increasingly decentralised assets and services. Performance-wise, the system remained stable and responsive under moderate load conditions

(up to 50 simultaneous market participants). The average block finality time and transaction throughput were within acceptable ranges for a local distribution market context. However, stress testing revealed that as transaction volumes increased beyond 100 concurrent events, the time to finalise contracts rose sharply. This highlights an important scalability constraint that must be addressed for real-world deployments at larger scales. One key insight was the framework's effectiveness at providing an immutable audit trail of all transactions, which could greatly reduce the administrative overhead and compliance burden faced by current market operators. Additionally, the use of open standards and smart contract modularity suggests the architecture could be adapted for different local flexibility market designs across regions or DNOs.

However, the results must also be viewed through the lens of their underlying assumptions. The simulation assumes full digital maturity of participating actors and uninterrupted data input from DERs and FPs. In practice, technical integration challenges—particularly with legacy DNO systems and smart metering infrastructure—may hinder seamless adoption. The analysis also presumes rational and honest behaviour from market participants, and does not yet model gaming strategies or adverse incentives that may emerge in live markets.

4.5 Conclusions

In conclusion, this chapter has successfully demonstrated the potential of integrating a ZK solution in flexible energy resources into a LV network, emphasising the crucial role of FPs with Solar Photovoltaic systems and Battery Storage systems. Through a detailed study, the work has designed an operational model for six FPs in a UK-based context, each operating two distinct types of assets that contribute to the network's overall energy flexibility and assessed how the ZK solution performs.

The model took into consideration real-world conditions, with the actual energy profiles reflecting varying sunlight availability and battery usage dynamics, providing a holistic view of how these assets perform on an average summer week. Utilising baseline energy profiles, calculated as an average of the preceding three weeks, the

work showcased the ability to contextualise the actual energy behaviour and performance, providing meaningful insights into the overall network operation.

Moving forward to the proposing a system of zero-knowledge proofs served as a robust method for verifying the correct operation of these FPs and ensuring fair settlements. By implementing these cryptographic protocols, the work enabled the DNO to validate the accuracy of the data and the compliance of flexibility delivery, without revealing sensitive information, fostering an environment of trust and transparency.

If the actual energy performance deviated significantly from the baseline, the system allowed for the imposition of penalties, promoting accountability amongst the FPs. On the other hand, successful delivery of the required flexibility led to fair remuneration, incentivising providers to optimally manage their assets.

S.No	Recommendations	Description
1	Evaluation of Cost and Efficiency	Perform a cost-benefit analysis considering both monetary and computational efficiency to compare the current system with the ZK solution.
2	Review of Data Security and Privacy	Assess the current system's data protection measures to quantify the added benefits of the ZK solution in terms of data security and privacy.
3	User Acceptance	Conduct surveys or interviews to understand the flexibility providers' acceptance of the new ZK system, identifying their concerns and opportunities for improving the user interface or experience.
4	Regulatory Compliance	Ensure the ZK solution aligns with existing energy regulations in the UK, or identify necessary regulatory modifications or allowances to facilitate the implementation of the ZK solution.
5	Scalability	Compare the scalability of the current system and the ZK solution. Assess how the ZK solution performs in a larger network scenario.
6	Transparency and Accountability	Evaluate if the increased transparency and accountability provided by the ZK solution can lead to better operational performance, higher compliance rates among providers, and increased overall trust in the system.
7	Testing and Validation	Execute a comprehensive testing and validation plan for the ZK solution, covering all possible scenarios such as erroneous data entries, partial data delivery, and varied flexibility delivery performance.

Table 4:6 Recommendations

Based on the findings of this work, several recommendations can be proposed for comparing the current flexibility delivery settlement with the zero-knowledge (ZK) solution for a smooth integration in the flexibility market:

- Evaluation of Cost and Efficiency: It is important to perform a detailed cost-benefit analysis comparing the current system with the proposed ZK solution. Although ZK proofs offer enhanced security and privacy, they may come with higher

computational costs. Therefore, a thorough analysis considering both the monetary and computational efficiency aspects is recommended.

- Review of Data Security and Privacy: As the ZK proofs inherently provide a superior level of data security and privacy, an assessment of the current system's data protection measures will be necessary. This can help to quantify the added benefits brought in by the ZK solution in terms of data security.
- User Acceptance: The transition from the current system to a ZK solution might be a significant shift for the FPs. Hence, it is recommended to assess their acceptance of the new system. Surveys or interviews could be conducted to understand their concerns and potentially improve the user interface or experience of the ZK solution.
- Regulatory Compliance: Regulatory factors play a critical role in energy markets. It is essential to ensure that the proposed ZK solution is in line with the existing energy regulations in the UK. If not, potential regulatory modifications or allowances should be considered to facilitate the implementation of the ZK solution.
- Scalability: The current system and the ZK solution should be compared in terms of their scalability. As the grid continues to evolve and more FPs are likely to be added, the system should be able to scale effectively. The ZK solution's performance in a larger network scenario should be assessed.
- Transparency and Accountability: The ZK solution is likely to increase transparency and accountability in flexibility delivery. A comparison should be made to evaluate if this can lead to better operational performance, higher compliance rates among providers, and increased overall trust in the system.
- Testing and Validation: A comprehensive testing and validation plan should be executed for the ZK solution, covering all possible scenarios, including erroneous data entries, partial data delivery, and varied flexibility delivery performance.

These recommendations are intended to ensure a holistic comparison and transition from the current flexibility delivery settlement system to a ZK solution, considering the different dimensions, stakeholders, and potential challenges involved.

The successful implementation of this work marks a significant advancement in managing the balance between energy supply and demand on the grid. It underscores the benefits of leveraging DERs, advanced data analytics, and cryptographic proofs in creating a more resilient, efficient, and sustainable power system. Furthermore, the insights gained from this work will undoubtedly contribute to the ongoing global efforts to integrate more renewable energy sources into the grid.

5 Decentralised ROSCAs for financing flexibility assets

This chapter proposes a blockchain-based ROSCA model to finance flexibility assets, thus allowing for a more customer centric energy transition. The motivation behind the work is to reduce barriers to entry for small-scale prosumers and promote collaborative investment in sustainable energy infrastructure. A smart contract architecture has been developed to manage contributions, auctions, and payouts without centralised oversight. The algorithm then adopted by a community-driven financing scenario to enable pooled investments in assets like solar panels or batteries, governed by transparent rules. A raffle and auction algorithm has been developed for participants to be able to over-pay on certain months, thus increasing their chance of winning the asset sooner. The MINA blockchain has been selected as the DLT architecture for this work due to its light-proof consensus mechanism, thus enabling fast transaction execution in a safe environment. A small energy community-based use-case has been developed for the adoption of 48 EVs, for which the algorithm has been tested on. The results indicate a potential saving of over £300,000 across the community for the duration of the scheme, when compared to a classic car leasing financial tool at a 10% APR.

5.1 Introduction

The drive towards sustainability and delivering Net Zero has become a key problem around the world, and in our days, it can be observed that the delivery of national decarbonisation strategies which dictate the pathway of reaching Net Zero by 2050 (“United Nations Framework Convention on Climate Change (UNFCCC,” 2015). If initially, many of these strategies adopted more of a technical standpoint in the show

of decarbonisation, recently it can be observed a transition towards the problem of governance and financing (Markard et al., 2012).

It is fair to assume that the penetration of renewable energy sources and storage at all points in the electricity network will imply that one of the most widely used tool of decarbonisation will be electrification (Lund and Mathiesen, 2009). Yet, considering the generation pattern and location of renewable energy sources, whole levels of system flexibility at the distribution network will be required to maximise their potential (Agency, 2019).

The desired level of flexibility will only be met if energy users will invest into generation, storage, and energy management (such as solar panels, EVs and energy management systems). The costs associated with these investments are impactful, and in the current economy, the customer will financially struggle to adapt (Sovacool and Griffiths, 2020). Moreover, the access to financial tools to offset the initial high capital expenditure of acquiring the above mentioned is minimal, with lack of government support in most countries (Brown et al., 2019).

One solution to the problem is the use of ROSCA as a novel way of financing the transition to Net Zero (Ardener, 1964). In the current environment, where the interest and work on energy communities and microgrids becomes more and more relevant, these communities can form ROSCA groups towards financially working together in delivering their role for Net Zero (Hicks and Ison, 2018).

Finally, the digital infrastructure of delivering and operating the ROSCA becomes a key challenge. As the trust is imperative for delivering Net Zero, the trust in this digital solution must be treated with an uttermost importance (Mayer et al., 1995). In (Swan, 2015), blockchain technology has been selected as the enabler of delivering this digital solution, due to its security, privacy, redundancy and trustworthiness characteristics.

5.2 Technical background

5.2.1 ROSCA

A ROSCA is a collective financial arrangement where individuals agree to meet for a specified period to save and lend money among themselves. This arrangement combines aspects of peer-to-peer banking and lending. Members contribute regularly to a common fund and take turns withdrawing the accumulated sums. Economist F.J.A. Bouman described ROSCAs as “the poor man’s bank”, (Bouman, 1979) highlighting how money is rapidly circulated, meeting both consumption and production needs.

ROSCAs are known by various names worldwide, some of which have become loanwords in different languages, including English, particularly in regional contexts (Ardener and Burman, 1995). For instance, in Latin America, ROSCAs are often referred to as “tandas”, among other names.

Meetings can be scheduled regularly or aligned with seasonal cash flow cycles in rural communities, often coinciding with crop harvests for farmers and pay dates for employed members when funds are readily available. Each meeting allows for one periodic money withdrawal, referred to as a slot. The order of money distribution is determined by drawing slots, which is agreed upon before starting the periodic fund accumulation. Members can mutually agree to swap slots based on their needs, provided this is done before the fund accumulation or withdrawal period. A member holding multiple slots may choose their preferred pay dates for these slots, but changes must be communicated to the organiser to avoid confusion. Each member contributes an equal amount at every meeting, and one member takes the entire sum, enabling access to a larger amount of money during the ROSCA's duration for personal use. This method is a popular alternative to saving at home, where funds may be accessible to family and relatives.

Transparency is a key feature, as all transactions are witnessed by every member during meetings. Since no money is retained within the group, record-keeping is minimal, often limited to a simple list of slots. This simplicity makes the system suitable for communities with low literacy levels and weak collective property rights

protections.

The time-limited nature of ROSCAs, typically lasting no more than six months, reduces risk for members. Each member receives at least one payout during the cycle, minimising potential losses if someone defaults early. ROSCAs thrive under two conditions that make them competitive financial alternatives, even in more sophisticated economies:

- The erosion of buying power of accumulated savings in inflationary conditions.
- The failure of conventional financing markets to provide credit to creditworthy borrowers due to opportunity costs, regulations, or operational expenses.

While the general mechanics of ROSCAs are simple, their continued success depends heavily on effective self-governance and collective enforcement. Trust and social accountability form the foundation of participation, but groups often develop informal rules, peer pressure mechanisms, or rotating leadership roles to ensure compliance and resolve disputes. In digital or large-scale ROSCAs, these governance functions may be supported by technology—such as smart contracts or automated reminders—to enhance reliability. This adaptability allows ROSCAs to maintain their core values while scaling into broader financial ecosystems, both online and in more formalised settings.

ROSCAs are informal or "pre-cooperative" microfinance groups documented extensively across the developing world. Anthropologist Clifford Geertz's early study of the arisans of Modjokuto in Eastern Java highlighted these groups. He described them as "an 'intermediate' institution emerging within the peasant social structure (Geertz, 1962), harmonising agrarian economic patterns with commercial ones, and acting as a bridge between peasant and trader attitudes toward money and its uses." Participants in a ROSCA select each other, ensuring that involvement is based on trust, social capital, and a genuine commitment to participate. In Brazilian consorcios, however, groups of strangers are brought together by an agent or intermediary, who facilitates group formation and administration for a fee. As of 2015, Brazil reported over five million active agricultural ROSCA users (Brazil, 2015). As the consorcio progresses, the same features of social capital and compliance

emerge, as members develop personal contact and trust.

Carlos Veléz-Ibáñez, an anthropology professor at Arizona State University, noted that technology has introduced a new dimension to savings pools, with "electronic cundinas" (ROSCAs) now being organised on websites that connect people across the United States (Vélez-Ibáñez, 2010). Some notable products include eMoneyPool, created by two brothers in Phoenix, Arizona; Monk, founded by ex-Google and ex-Intel employees in Silicon Valley; Puddle, a Google-venture backed startup; Moneyfellows, a GB and African-based platform digitising the ROSCA model; ROSCA Finance, a global, autonomous money-sharing platform founded by former Santander bankers; Esusu, started by ex-Goldman Sachs, PwC, and LinkedIn employees in New York; and Partnerhand, a UK-based organisation facilitating online 'Pardner's' among verified individuals, founded in 2010.

StepLadder, founded in 2016 by finance professionals with significant academic work on Consorcios in Brazil, has entered the GB market, targeting prospective first-time home buyers with ROSCA-based collaborative finance. In October 2017, the Finlok platform launched a digital ROSCA product in India, leveraging NPCI's Unified Payment Interface.

Aturi Africa has automated and digitised Chama financial services, aiming to provide these services to millions of people in Africa and globally. The FinTech startup, founded by a former Safaricom employee from Kenya, launched in late 2020.

In 2022, a more comprehensive version of a savings club for purchasing vehicles was launched in the United States. “savings.club” allows users to join clubs administered by the company, offering rates significantly lower than traditional auto loans by leveraging credit card payments.

5.2.2 Blockchain-based ROSCAs

As mentioned previously, trust is essential for traditional ROSCAs, where members depend on each other's reliability and integrity to meet financial commitments. This trust is built through personal relationships, social accountability, and visible transactions during meetings. However, moving ROSCAs to a digital platform poses

a challenge in replicating this level of trust. Participants no longer interact face-to-face, reducing the social pressure and personal connections that ensure compliance. In digital implementations, an even higher degree of trust and transparency is necessary, as if the ROSCA is set up by using a random draw for distributing the acquired items at the beginning of each period, the participants must trust that the draw has been conducted trustfully.

Blockchain technology is well-suited to address this need due to its decentralised and immutable nature. Every transaction in a blockchain-based ROSCA is recorded on a public ledger, visible to all participants. This transparency ensures that all business logic such as transactions and withdrawals are verifiable, reducing the risk of fraud and default.

Smart contracts further enhance trust by automating fund collection and distribution based on predefined rules. These self-executing contracts remove the need for a central authority or intermediary, minimising the risk of human error or manipulation. They can manage various contingencies, ensuring funds are only released when conditions are met, maintaining the integrity of the ROSCA.

Additionally, blockchain's security features protect against unauthorised access and tampering. Each transaction is encrypted and linked to the previous one, making it nearly impossible to alter records without detection. This high level of security reassures participants that their funds are safe, and that the system is resistant to breaches.

By leveraging blockchain, digital ROSCAs can create a trust environment similar to traditional setups, where accountability is maintained through transparent, automated, and secure processes. This technological foundation not only preserves the core principles of ROSCAs but also enhances them, making digital ROSCAs a strong alternative for modern financial inclusion.

On addition, blockchain-based zero-knowledge proofs (ZKPs) are highly beneficial for ROSCAs because they enhance privacy, security, and trust without sacrificing verifiability. ZKPs allow one party to prove the validity of a transaction without

revealing specific details. This means participants can verify contributions and withdrawals (such as soft credit checks) in a ROSCA without exposing sensitive financial information (Ben-Sasson, 2014).

In a blockchain-based ROSCA, ZKPs ensure all transactions are legitimate and follow the group's rules while keeping individual contributions and withdrawals confidential. This privacy is important for maintaining trust, as it protects members' financial details and prevents misuse of personal data.

5.2.3 Financing for flexibility

The interlinked threats of climate change and biodiversity decline are increasingly being felt all over the world. From extreme summers, such as the constantly appearing heat waves in the UK, to devastating floods in Pakistan (Bank, 2022) and the loss and fragmentation of habitats globally, the imperative to transition to Net Zero and adapt to climate change is becoming increasingly acute.

Yet most of the forecasted investments are focusing on providing the required incentives towards shifting large generation power plants towards renewable energy sources (such as large solar farms or offshore wind), or towards budling or upgrading the required electricity network infrastructure to deliver the energy transformation. Or, in some cases, by allowing the generation of green certificates for investment in low-carbon generation.

Recently, among European countries (including UK) it can be observed a shift of interest towards local demand side flexibility as one of the primary tools of delivering Net Zero. Flexibility is crucial to operating the energy system where the supply and demand of energy needs to be balanced over different timescales.

Operating a future energy system with high levels of renewable energy and no unabated natural gas generation will require much more flexible, zero-carbon capacity than is available today. New technologies will need to provide the services that natural gas has traditionally supplied. System stress will increasingly come from fluctuations in electricity supply and demand, not just peak demand. Significant short-term flexibility will be needed to balance supply and demand within a single

day. During periods of high or low renewable generation, even more within-day flexibility will be necessary. Extreme weather events will require flexibility over weeks, necessitating a diverse set of flexible solutions to ensure the system's reliability during these times. The expansion of distributed flexibility—such as storage, EVs, heat pumps, and thermal storage at the distribution level—is essential for achieving Net Zero.

UK's ESO Demand Flexibility Service has shown that consumers are willing to participate in demand-side response (DSR), but this is just the starting point. Effective future DSR will need suitable market signals and technological improvements. GB ESO's Net Zero scenarios predict that DSR potential could reach 6-12 GW by 2040 from the residential, commercial, and industrial sectors, as presented in Figure 5:1.

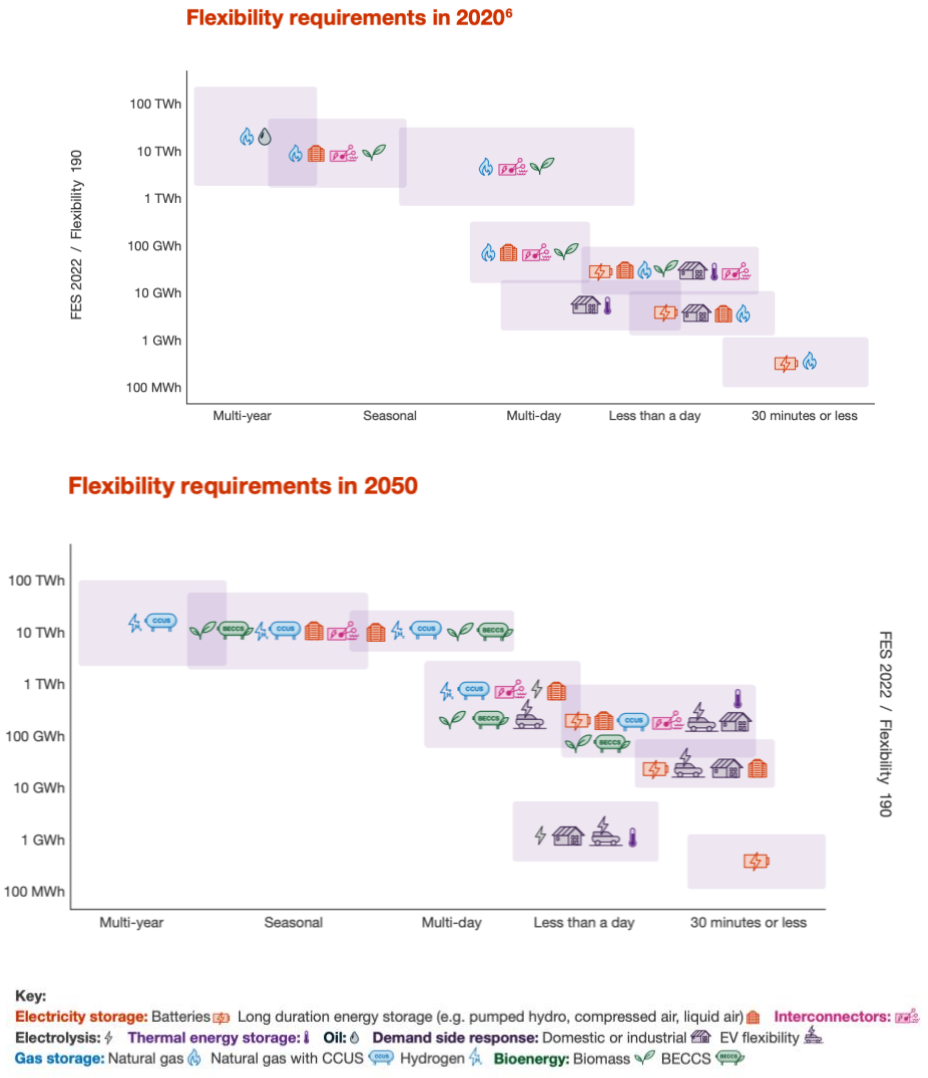


Figure 5:1 Flexibility requirements (ESO, 2021)

Accessing these challenges requires significant upfront public and private capital. Between 2023 and 2050, £120 trillion of total investment in the energy transition will be required for the world to align with a 1.5-degree pathway (I.R.E.N.A., 2019). The is an incontestable gap in the required level of public and private investment, as this required annual investments to quadruple from current levels (McKinsey and Company, 2022).

For a full commitment of participation in flexibility (Business et al., 2020), the consumer must have access to electricity generation and storage, thus for a minimum, and in the current context, solar PV generation, an EV, and driven by the requirement of decarbonisation of heat, a heat-pump.

The cost of a 4kW solar panel system, which is common for households, ranges from £6,000 to £7,000. This includes around 12-16 panels, depending on their efficiency and wattage. Installation costs can add an additional £600 to £3,000 depending on the complexity of the installation. The price of an EV in the GB varies widely depending on the model and specifications. On average, a mid-range EV like the Nissan Leaf costs around £28,000 to £32,000, while premium models like the Tesla Model 3 can range from £40,000 to £50,000.

The cost of installing a heat pump for an average household can range from £2,400 to £14,400. This includes the price of the heat pump unit itself, which varies based on the type and capacity, and installation costs, which can differ based on the complexity of the setup and specific household requirements.

This leads to a total average required cost of over £40,000, which will be extremely difficult for a household to put towards in the current state of the economy (Trust, 2023). Different governments have put in place different schemes for providing incentives or grants towards helping customers. UK's Green Deal helps households make energy saving improvements to their home and help them find the best way to pay for them. Octopus Energy, a GB electricity supplier, installs heat pumps and helps their customer applying for the UK's heat pump grant, which is up to £7,500.

Yet the availability of these grants is based on specific requirements, which might not

all the time allow the household a good enough level of flexibility of choice. Additionally, it will be difficult for DNOs to fully maximise the upcoming levels of flexibility is they have no transparency over how much level of increase in customer flexibility, and over what period of time it will be made available.

5.2.4 The concept of energy community

According to the EU, and inherited in UK, energy directives (Parliament and European Union, 2019), citizen energy communities are legal entities that allow citizens to engage directly in energy consumption, generation, storage, and trading activities. These communities have the right to access suitable markets either as self-representing entities or by hiring a third-party energy service provider, such as an aggregator. Energy communities are often structured as cooperatives or municipal corporations and can evolve into various legal forms. Their primary focus is on providing affordable energy to their members or shareholders and facilitating the adoption of new technologies.

Increasingly, citizens are becoming 'prosumers,' pooling their resources through these initiatives and enabling community participation in electricity markets. The goal is to empower consumers to manage their energy mix behind the meter (BTM), reducing costs while enhancing grid flexibility and security. However, the development of Local Energy Communities brings technical challenges and necessitates new social, economic, and regulatory arrangements.

Energy communities promote local sustainable energy production, create new business opportunities, and form new types of energy providers while interacting with local, regional, and national power systems, thereby bringing social innovations in a decentralised energy system that can operate independently. These communities encourage end-users to utilise various energy technologies to optimise their energy costs and support local economic growth by trading flexibility services in emerging electricity markets. Typical generation technologies in Local Energy Communities include photovoltaic systems, wind turbines, and Combined Heat and Power plants. Additionally, thermal energy systems, solar heating, and heat pumps are employed to meet heat demand.

Storage technologies, including Battery Energy Storage Systems and thermal storage, are utilised to effectively manage local renewable energy production. The flexibility provided by these storage systems, along with heat pumps, EVs, and water boilers at the household level or as shared assets within local energy communities is crucial for increasing energy efficiency and minimising costs for end-users. The emergence of these communities has created economic and sustainable opportunities for DSOs to deploy their flexibility potential for various grid services. Aggregating lower capacity assets within a community can diversify the range of flexibility services and reduce risks for service providers.

Aggregators play a major role in procuring flexibility from the energy communities and trading it to responsible parties, DSOs, or transmission system operators. This flexibility chain involves the interaction of many stakeholders, where participation depends on various factors, including economic benefits. Traditional power systems are well-established with institutional and industry actors covering generation, transmission, and distribution, forming efficient socio-economic alliances. The technological and social innovations brought by these local energy communities require cooperation from key actors to achieve their objectives, and technology actors should collaborate with the community and other stakeholders to ensure success (Caramizaru and Uihlein, 2020). USEF conceptualises this in Figure 5:2.

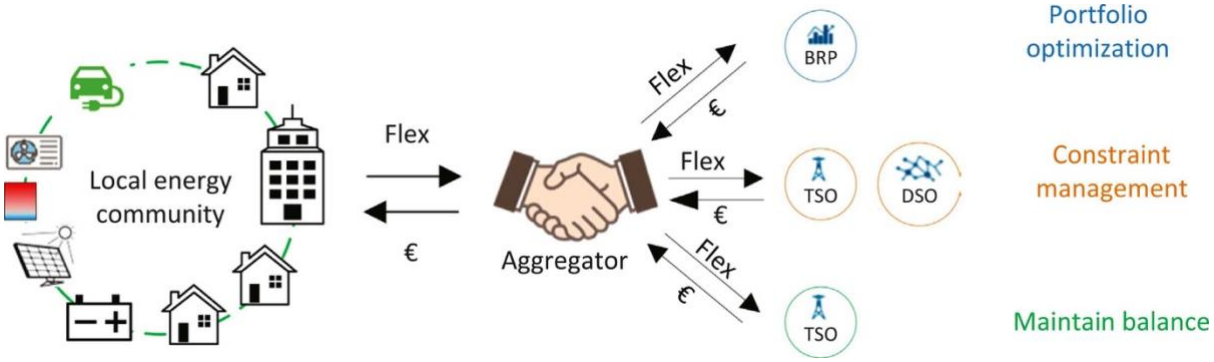


Figure 5:2 USEF local energy community

5.3 Methodology

5.3.1 ROSCA Design

In the context of empowering the prosumers and flexibility assets, this work proposes the use of ROSCAs as the main delivery tool in supporting the energy community

users with access to flexibility asset. This work plans to challenge the way energy communities interact with using financial products for their purchases. The core logic is broken down into three stages:

- the discovery stage,
- the progress stage, and
- the settlement stage

A retailer, goods dealer or supplier identifies a product that they acknowledge a large group of people might want, and that can provide (an EV, a heat pump, a PV, etc). The retailer then takes the role of a Group Owner (GO). The GO will then form a group and allow users to express their interest in joining. Once the users have joined the group, the collective group decides on the group settings (payment size and timeline). Once the decision has been taken, the GO will confirm the settings and lock the settings and the group. The users will then have to confirm their acceptance of the proposed settings. If the settings are not approved, this becomes an iterative process. If the settings are confirmed, the group then is formed, and the “Discovery” process is finished. This is illustrated in Figure 5:3.

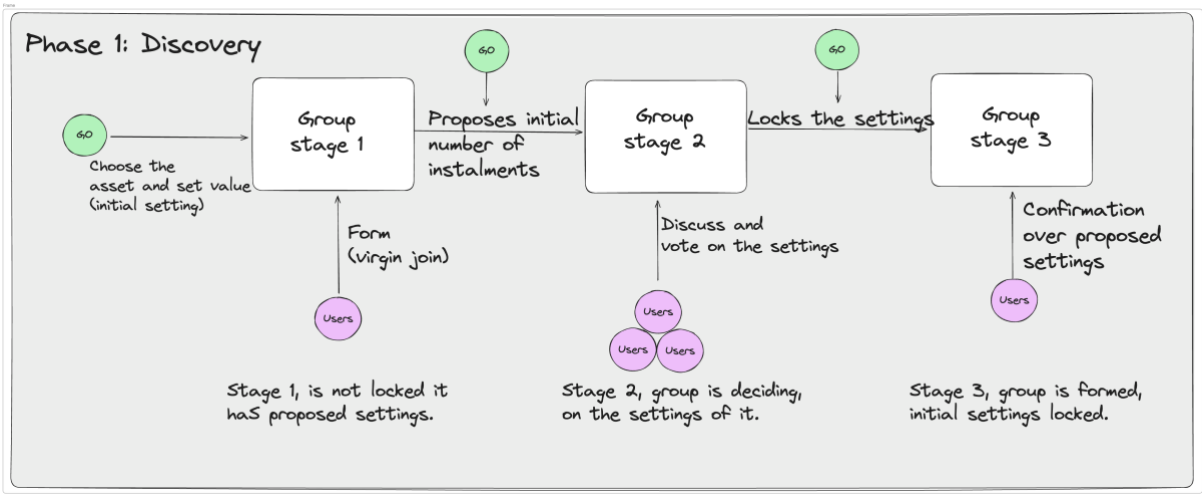


Figure 5:3 Discovery phase

The second part of the process is the “Progress” part, as illustrated in Figure 5:4. Each month, users pay the instalment, which will allow them to purchase 2 of the goods. This is because two processes are happening every month: firstly, one winner will win the first good purchased. The second good will be delegated to the

user that wins the auction process. The auction process requires willing users to pay a number of instalments in advance. The more instalments they pay in advance, the more chances they have to win the second asset. The second good will be awarded to the user who paid the highest number of instalments in advance. Both of these processes happen monthly. The benefit of this model arises from the building up of liquidity to be able to address potential risks occurring in the future and to incentivise users to pay earlier.

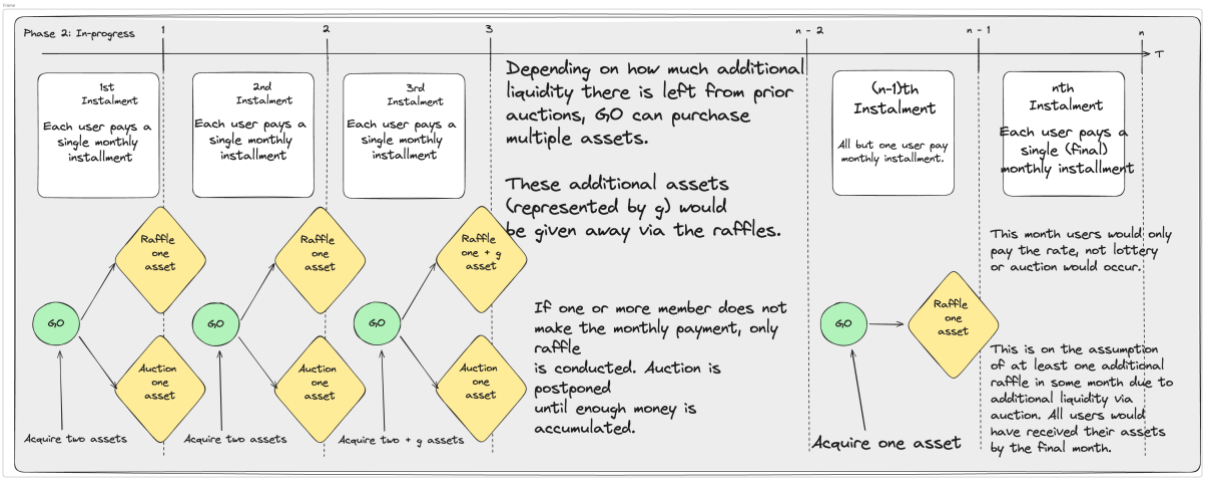


Figure 5:4 Progress phase

The following can be illustrated with this example:

- The group item, an EV, has a price of £24,000. The initial duration of the group is 48 months, and there are 96 members in the group.
- Every month, each member contributes £500, resulting in a total monthly collection of £48,000, sufficient to purchase two EVs. One EV is awarded through a raffle.
- For the auction, members have the option to bid additional instalment payments. For instance, in the first month, User A wins the auction by offering an extra 5 instalments. Upon payment, User A receives the car, but they must continue making regular monthly payments as required. This user has now 5 fewer instalments to pay, therefore he finishes 5 months earlier.
- In subsequent months, the process repeats. If there's no auction bid in a month, both cars are raffled. If there are bids, the second car goes to the highest bidder.
  • Accumulated extra payments from auction winners lead to the group having enough funds for an additional asset, further increasing the efficiency of the

model, and shortening the time window for receiving the asset.

It is important to note that for each user that won the product before the period is finished, the user leases the product from the GO. Therefore, the users will only fully own the product at the end of the final instalment, when the group terminates.

Figures 5:5 to 5:8 provide a structured visual explanation of how the asset distribution mechanism operates under the proposed rotating savings and credit model, comparing scenarios with and without auction-based prioritisation. Figure 5:5 depicts the base case in which no members participate in the auction mechanism. In this scenario, all users contribute equally throughout the group duration, and assets are allocated solely through a monthly raffle. As a result, every user receives an asset by the end of the fixed term, maintaining a uniform structure of payments and delivery.

Figure 5:6 introduces the auction component into the system. Here, users can bid additional instalments in advance in order to increase their likelihood of receiving an asset earlier. This mechanism enables two assets to be distributed each month: one through the existing raffle system and one awarded to the highest bidder in that period. Importantly, the user who wins through the auction receives the product earlier but continues paying the regular monthly instalments, with the extra upfront payments reducing their overall payment duration. Figures 5:7 and 5:8 provide further clarity by illustrating individual outcomes in specific months. In Figure 5:7, during Month 2, the asset allocation shows one asset delivered to a raffle winner and the second asset allocated to a user who made the highest auction bid, offering five additional instalments.

This results in an earlier receipt of the asset and a shortened commitment period. Figure 5:8 demonstrates a case in Month 4 where two users place identical highest bids. In this case, the tie is resolved through a fair selection process, such as a random draw between the tied bidders.

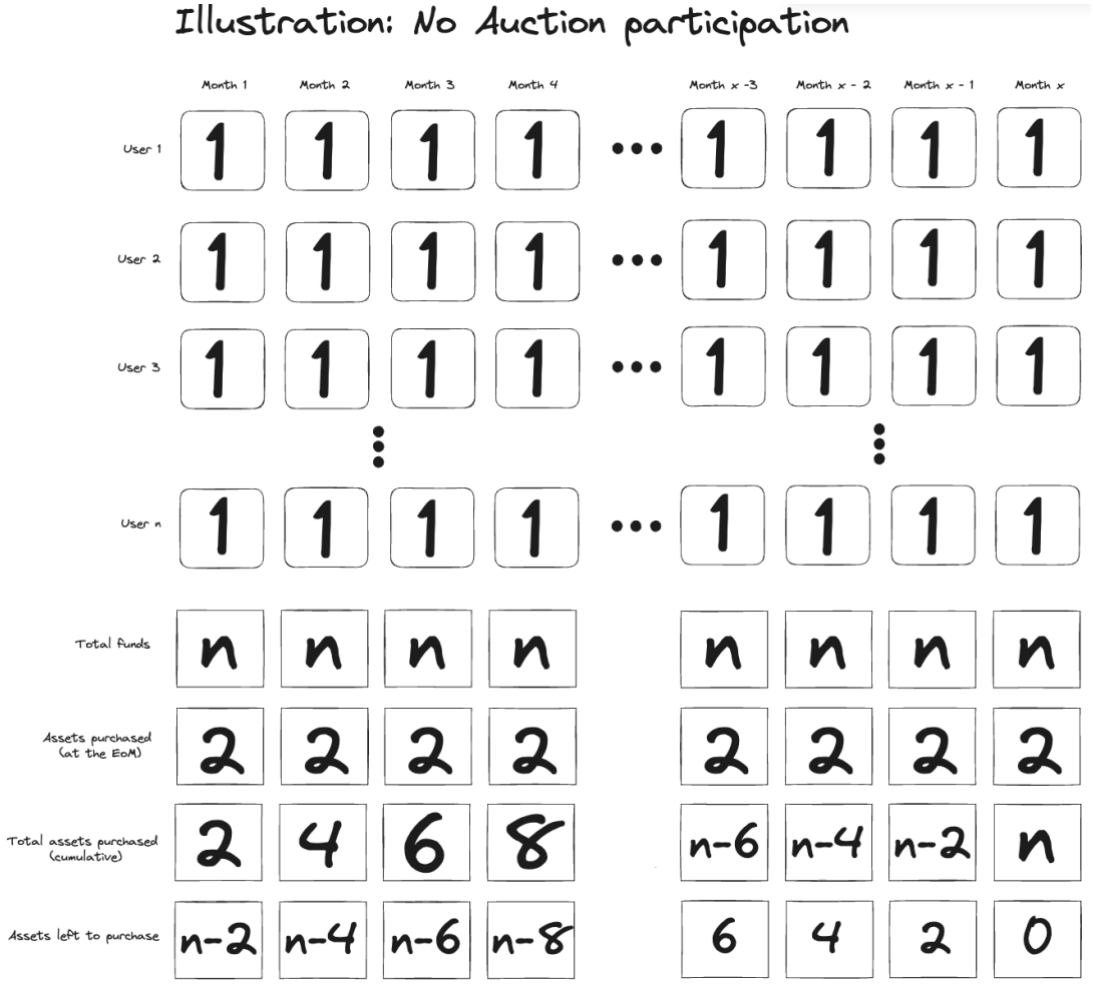


Figure 5:5 ROSCA without auction participation

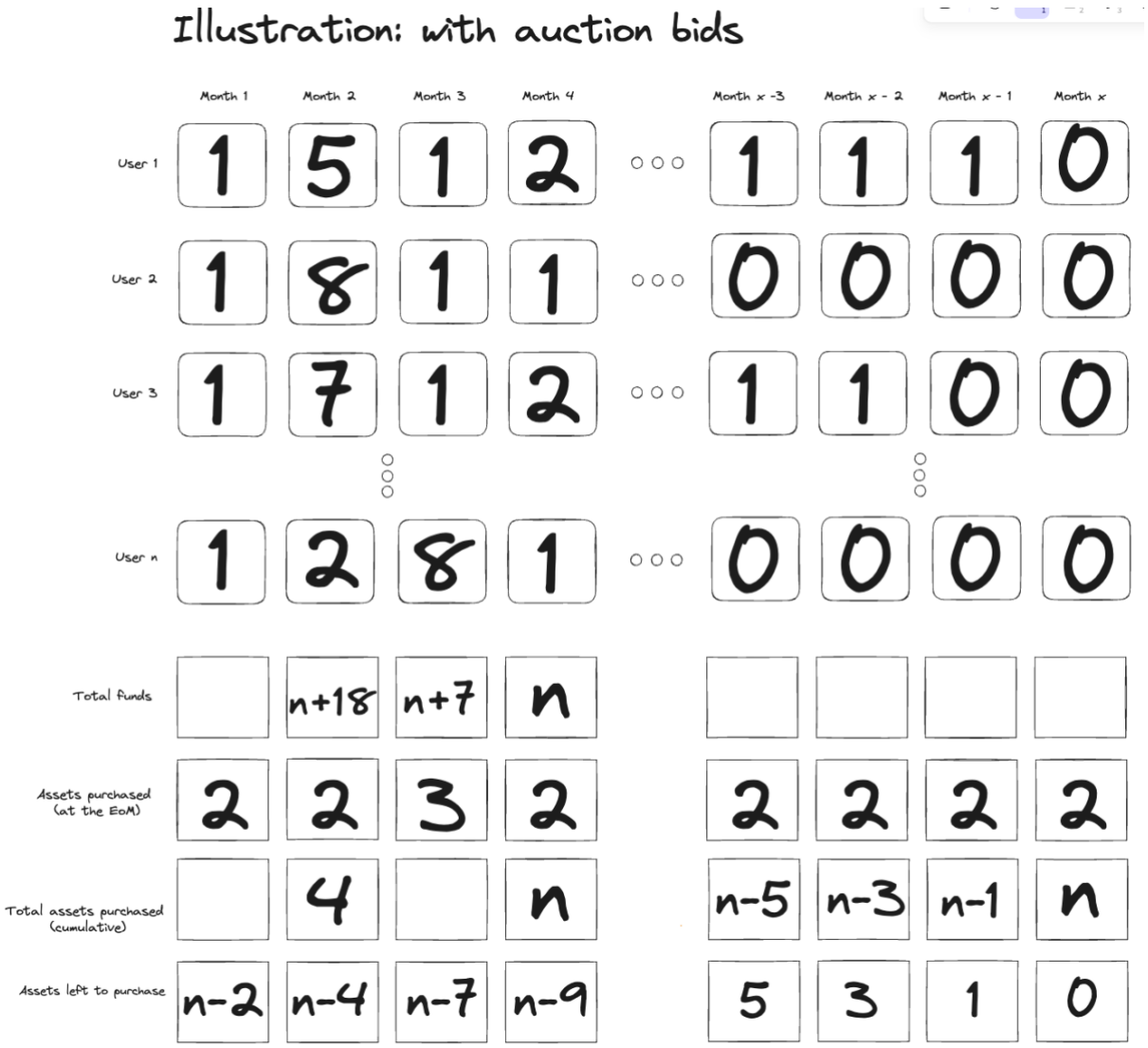


Figure 5:6 ROSCA with auction participation

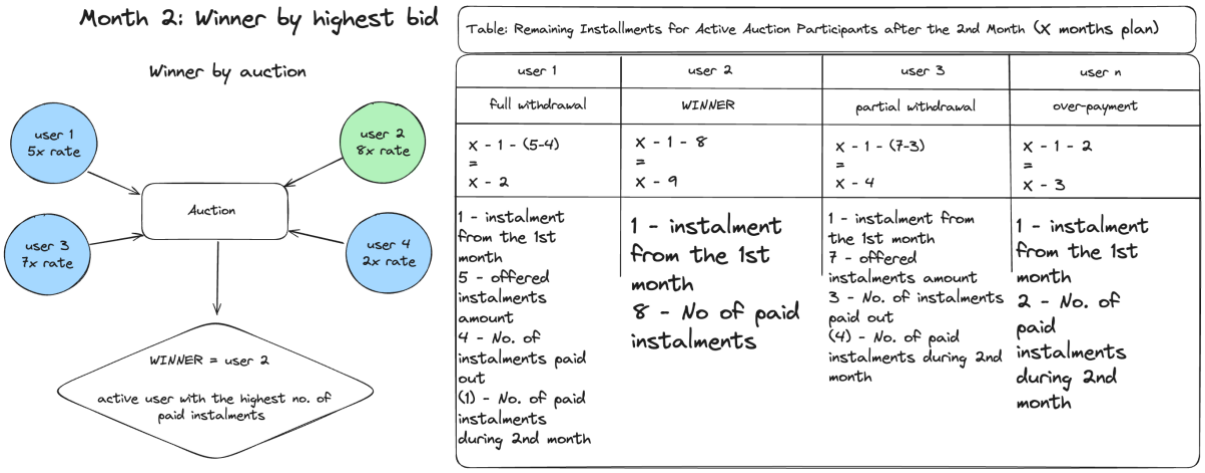


Figure 5:7 Month 2 illustrating the winner by the highest bid

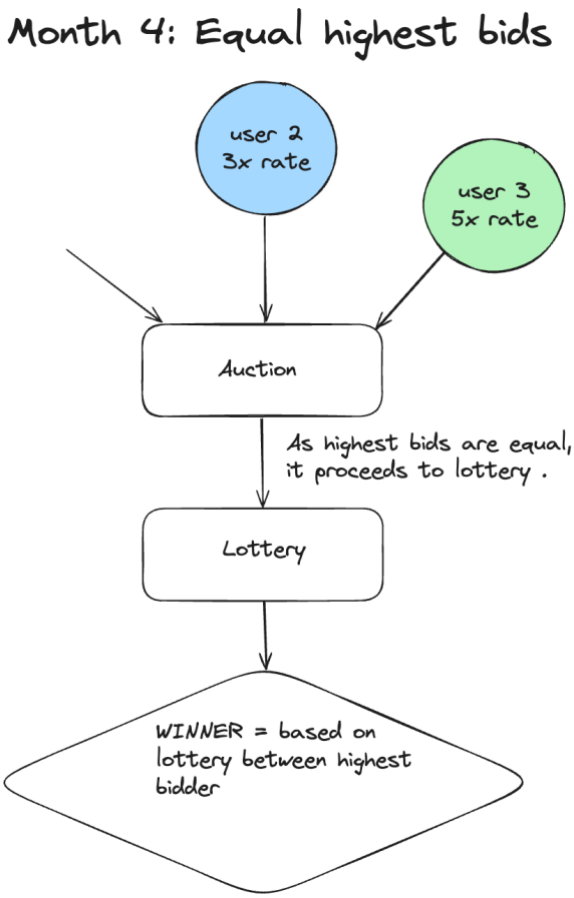


Figure 5:8 Month 4 illustrating equal highest bids

As showed in Figure 5:9, the third part is the “Settlement” part where the ownership over all the assets moves from the GO to the users.

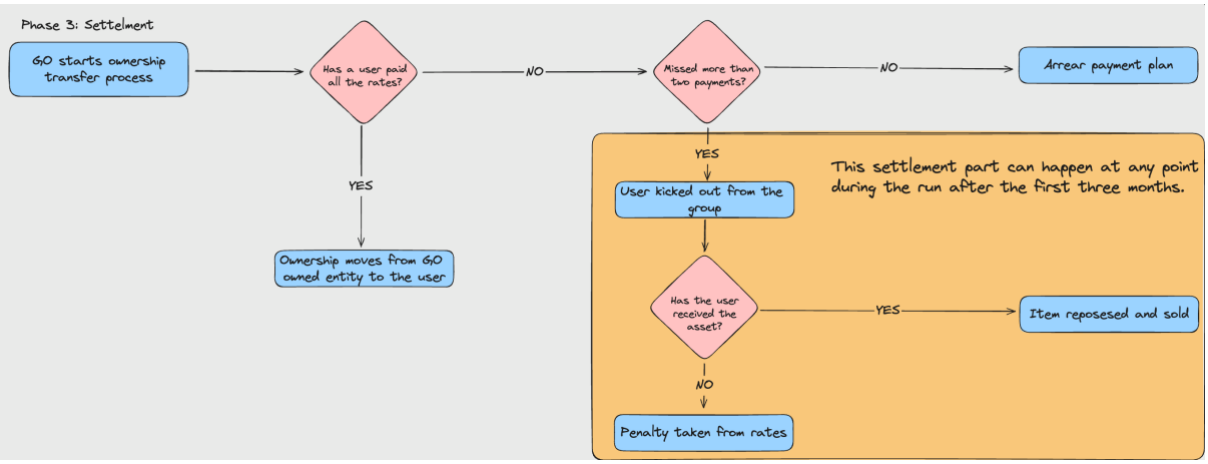


Figure 5:9 Settlement process

Multiple groups can be formed at the same time, and one player can have multiple

roles at the same time for different groups. For decisions that require member voting, such as electing group leaders or making significant group purchases, Mina's ZKPs will ensure that votes are counted without revealing individual choices. This maintains the integrity of the voting process while keeping individual preferences private. Each group will then act as its sole DAO, with the functionality to vote and agree on decisions.

An admin creates a contract for each group, and users can request to join. The admin must add each user to the group. Once the group is full, it begins. Users can pay in instalments, and at the end, the admin calls the Verifiable Randomising Function (VRF) to select winners. Winners can claim their prize, and the admin receives the money. The cycle repeats until the group is started again.

5.3.2 Auction and raffle mechanism

The auction model in the blockchain-based ROSCA system is structured as a sealed-bid first-price auction, where participants bid additional contributions to win a valuable asset (such as an EV) earlier than through the regular rotation. Each participant has the opportunity to bid an amount above their fixed monthly contribution, and the participant who submits the highest bid wins the asset. This model incentivises participants who are eager to receive the asset early to bid more, while ensuring fairness by giving all participants an equal opportunity to participate in the auction.

Let N denote the total number of participants in the ROSCA group. Each participant is required to make a fixed monthly contribution C. This contribution is uniform for all participants and is part of their commitment to the ROSCA. Each participant i can place an additional bid B_i on top of their monthly contribution C The bid B_i represents the extra amount the participant is willing to pay to win the asset early.The value of the asset being auctioned, such as an EV, is denoted by V. This is the same for all participants. The auction seeks to allocate the asset to the participant with the highest bid. The winner W is determined as the participant whose bid B_i is the highest:

$$W = \arg \max_{i \in \{1,2,\ldots,N\}} B_i \quad \textbf{(21)}$$

The participant with the highest bid wins the asset for that month. In a first-price auction, the winner must pay the exact amount of their bid.

Once the winner is determined, their total payment consists of their fixed monthly contribution C plus their winning bid B_{max}, where B_{max} is the highest bid. Therefore, the total payment for the winner is:

$$Payment\ for\ the\ winner = C + B_{max} \quad \textbf{(22)}$$

All other participants only pay their regular monthly contribution C and do not incur any additional costs. The funds from the winning bid are added to the ROSCA pool, increasing the collective liquidity. This mechanism incentivises participants who wish to receive the asset early to place higher bids, while those willing to wait for future rounds only make their base contributions.

The raffle model in the ROSCA system is designed to allocate an asset to a participant randomly each month. Unlike the auction, where participants compete by placing bids, the raffle offers an equal chance for each eligible participant to win the asset, ensuring fairness and providing an opportunity for all participants to benefit from the ROSCA regardless of their financial capacity to bid in the auction. The raffle incentivises regular participation and contributions since only those who have contributed their fixed amount for the month are eligible to win the asset.

Let N represent the total number of participants in the ROSCA group. Each participant has an equal chance of winning the raffle. As in the auction, each participant is required to make a fixed monthly contribution C. Only those participants who have contributed C for the current month are eligible for the raffle. Since the raffle is based on random selection, each participant i has an equal probability of winning. The probability pi that a given participant wins the raffle is uniform and is given by:

$$P_i = \frac{1}{N} \tag{23}$$

Thus, each participant has a $\frac{1}{N}$ chance of winning the asset in any given month, provided they have made their required contribution. The raffle winner W_{winner} is determined by a random selection process, which can be implemented using a decentralised random number generator (Chainlink VRF for verifiable randomness in the blockchain system). The winner is selected from the pool of eligible participants:

$$W_{winner} = Random\ (1, N) \tag{24}$$

This ensures that the selection process is transparent and tamper-proof. Since the raffle is a random event, each participant's expected value of winning the asset in any given month is:

$$E[Win] = \frac{V}{N} \tag{25}$$

where V is the value of the asset. This expected value represents the average benefit that each participant could expect over time if the raffle were repeated across multiple cycles.

The raffle winner does not pay any additional amount beyond their regular contribution C. Once selected, the asset is transferred to the winner, and the ROSCA continues into the next cycle. This mechanism ensures that even participants who may not be able to bid in the auction have a fair opportunity to receive the asset.

The raffle provides a balance to the auction system, ensuring that participants with lower financial means are not disadvantaged, and it maintains the inclusiveness of the ROSCA by offering an equal chance to all.

5.3.3 Smart contract implementation

The blockchain architecture for the proposed energy community ROSCA (integrates smart contracts to automate and decentralise financial management, asset distribution, and energy flexibility trading. This system leverages blockchain's core

benefits—transparency, immutability, and security—allowing prosumers to manage contributions, participate in auctions, and receive assets like EVs or heat pumps without relying on traditional intermediaries. Central to the architecture is a layered system of smart contracts that governs group formation, contributions, raffles, auctions, and energy transactions, while ensuring fairness and privacy through cryptographic methods like ZKPs. By integrating blockchain with real-world oracles and energy management platforms, this architecture not only empowers prosumers financially but also optimises energy usage and flexibility within the community, enhancing both individual and collective sustainability.

5.3.3.1 GO Contract

The GO smart contract serves as the foundational layer for managing the ROSCA cycle within the blockchain-based energy community. Its primary role is to facilitate the organisation of groups, allowing participants to contribute towards the acquisition of assets such as EVs, heat pumps, or other flexibility assets. The contract ensures transparency, fairness, and automation in the management of contributions, raffles, auctions, and settlements. By defining the rules for asset acquisition, participation, and settlement, the GO contract eliminates the need for a centralised authority and promotes trust within the community. The contract handles the registration of participants, the management of contributions, the organisation of raffles, auctions, and the final asset distribution at the end of the ROSCA cycle.

The first step in the ROSCA process involves registering participants who want to join the group. The GO contract manages this by setting up a list of eligible members and verifying their identities through blockchain wallets. Each member must agree to the terms of the ROSCA, including the contribution amount, duration, and asset type before being admitted into the group. The contract creates a registration window where participants can sign up. Once the maximum group size is reached or the window closes, no further participants can join. Each participant's wallet address is recorded, and their commitment to the ROSCA cycle is secured.

The contract allows the GO to define the asset for which the participants will be contributing. This asset could be an EV, heat pump, or any other energy-related asset. The asset type is predetermined by the GO and communicated to the

participants during the registration phase. Upon group creation, the GO sets the asset type and its value. The contract records this information to ensure that contributions are correctly directed toward the purchase of this specific asset.

The duration of the ROSCA cycle determines how long participants will be contributing and when they will be eligible to receive the asset. The GO contract enforces these rules, specifying the contribution frequency (e.g., monthly), the total number of cycles, and the conditions for ending the ROSCA. The smart contract locks in the rules governing the cycle, such as the total number of participants, monthly contribution amounts, and the length of time each participant must contribute. It ensures that no participant can withdraw or stop contributing prematurely.

The contract automates the contribution process, ensuring that each participant pays their dues on time. It also manages the selection of asset recipients via a raffle system for the first good and an auction system for the second. At the end of the cycle, the contract facilitates the transfer of ownership for all assets. Contributions are collected periodically, and the contract tracks whether participants have fulfilled their payments. Each month, one participant is chosen via a random raffle to receive an asset, and another through an auction where participants can bid extra payments. Settlements are processed at the end of the cycle, ensuring all participants either receive the asset or are refunded their contributions if the group ends. The functionality of the smart contract is presented in Figure 5:10. The figure illustrates the structural relationship between two core entities: the contract held by the GO and the Participants. The GO contract stores global parameters for the ROSCA cycle, such as the number of participants, contribution amount, total contributions, asset type, and the current cycle status. It includes mappings to track each participant’s details and outcomes of the monthly raffles and auctions. The contract also contains functions to handle registration, payment contributions, winner selection, and asset distribution. Each Participant entity records wallet address, contribution history, and their status in the raffle and auction processes. The design ensures that logic and data are clearly separated and securely handled, allowing transparent and verifiable asset allocation throughout the cycle.

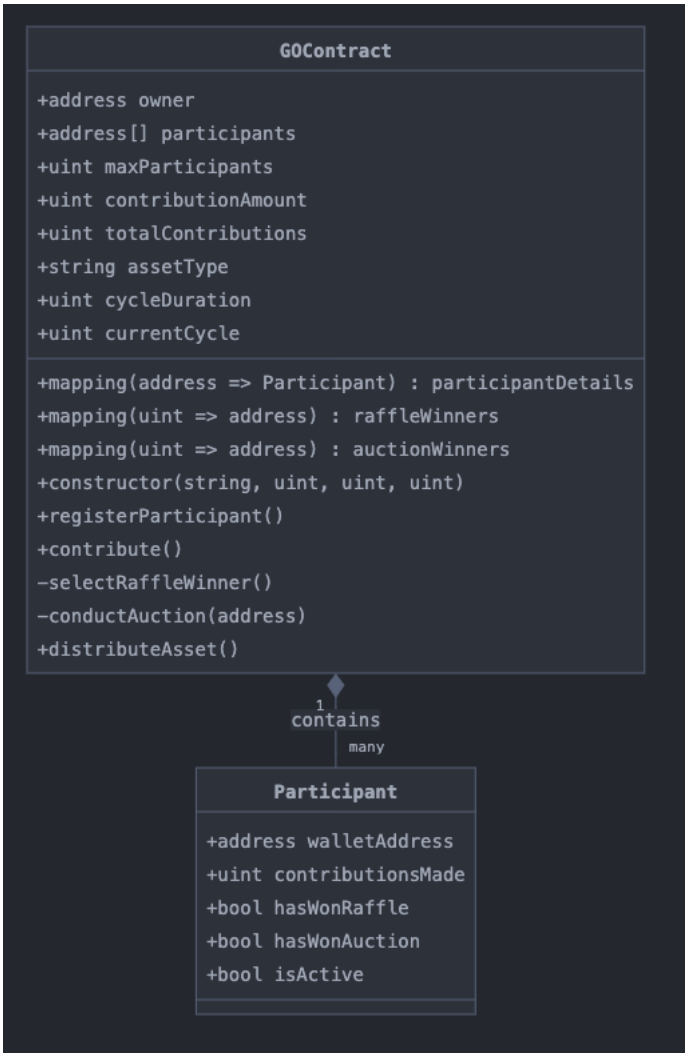


Figure 5:10 GO smart contract

5.3.3.2 *Contribution Management Contract*

The Contribution Management Contract is responsible for handling the financial transactions within the ROSCA. Its primary function is to ensure that all participants make their monthly contributions, track those contributions securely, and only release funds during specific events like auctions or asset distributions. By locking the contributions within the smart contract, participants are assured that funds are safe and inaccessible until they are properly allocated. This contract ensures transparency in tracking who has contributed and automates the management of payments, reducing the need for manual oversight while providing verifiable records of all transactions.

The core task of this contract is to manage and enforce monthly contributions from all participants in the ROSCA. It ensures that participants pay the correct amount at regular intervals, tracks due dates, and penalises late or missed contributions. The contract sets a recurring deadline by which each participant must submit their monthly contribution. If a participant fails to contribute, they may face a penalty, such as being temporarily barred from winning in the next auction or raffle.

All contributions made by participants are locked within the smart contract to prevent premature withdrawals or misuse of funds. These funds are only released when a participant wins an auction or raffle, ensuring the financial security of the group. Upon receiving a contribution, the contract automatically locks the funds in a secure escrow-like system, preventing any access until a trigger event (e.g., winning a raffle or auction) occurs. The funds are safeguarded by the smart contract's internal logic and are only released according to predefined rules.

The contract keeps a detailed record of each participant's contributions, ensuring that the group can see who has contributed and how much. This prevents discrepancies or disputes about payments, as the smart contract provides a transparent, immutable ledger of all transactions. The contract maintains a mapping that tracks how much each participant has contributed over time. This information is stored on-chain, ensuring it is tamper-proof and available for auditing by any participant.

Funds can only be withdrawn by participants in certain situations, such as when they win an auction or receive an asset. This ensures that participants can't prematurely access funds but are rewarded at the correct time based on the ROSCA rules. The contract checks specific conditions before allowing a withdrawal. If the conditions are met (e.g., the participant wins the auction or raffle), the contract releases the correct amount of funds. Otherwise, it denies any attempt to withdraw. The functionality of the smart contract is presented in Figure 5:11.

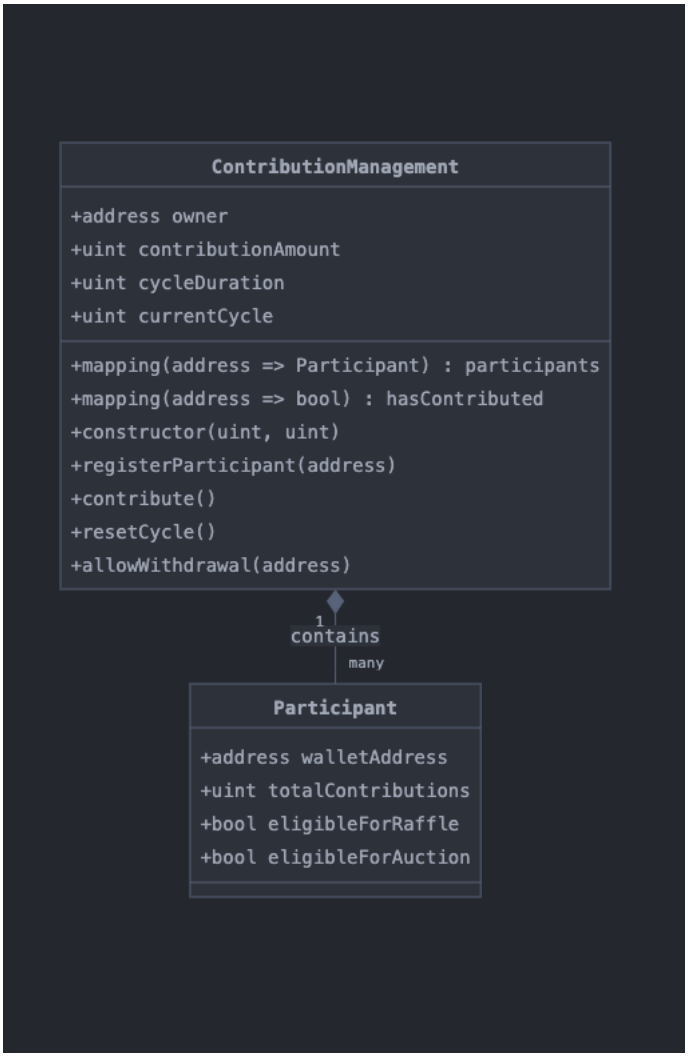


Figure 5:11 Contribution management smart contract

5.3.3.3 Auction Smart Contract

The Auction Smart Contract manages the auction process within the ROSCA cycle, where participants bid for the chance to receive an asset (e.g., an EV, heat pump, etc.) by offering extra contributions or instalments upfront. The contract ensures that the auction is conducted fairly and transparently, automatically selecting the highest bidder and handling the transfer of the asset. The contract collects bids from participants, compares them, and determines the winner at the end of each cycle. It ensures that once a bid is placed, it cannot be reversed, and the funds are locked until the auction concludes. This process incentivises participants to bid higher to receive the asset earlier, while also maintaining the integrity and transparency of the auction process on the blockchain.

The contract allows participants to submit bids, which represent either additional contributions or future installment payments made upfront. This bidding process occurs within a defined time window for each ROSCA cycle. During the auction period, participants call the contract's bidding function and submit their bids. The contract records the bid amount and participant details, ensuring the bid is valid (i.e., the participant has contributed the required amount for the cycle).
At the end of the auction period, the smart contract compares all the submitted bids and automatically selects the participant with the highest bid. The winner is granted the asset for that ROSCA cycle, while other participants retain their contributions for future cycles. Once the auction closes, the contract iterates through the bids, determines the highest one, and assigns the asset to the winning participant. This process is automated and transparent, ensuring fairness.

All bids placed in the auction are locked in the smart contract until the auction concludes. This prevents participants from withdrawing or altering their bids after submission, ensuring the integrity of the auction process. Once a participant places a bid, the contract locks the additional funds or upfront payments within the contract. These funds are only released once the auction is over and the asset is allocated.

After determining the highest bidder, the contract manages the transfer of the asset to the winner. For physical assets, this could involve interacting with a third-party oracle to confirm delivery, while for digital assets (e.g., energy credits), the transfer can happen directly on-chain. The contract triggers the asset transfer function once the auction ends. If a physical asset is involved, the contract works with an oracle to ensure the asset is delivered. If it’s a digital asset, the ownership is transferred directly via the blockchain. The functionality of the smart contract is presented in Figure 5:12.

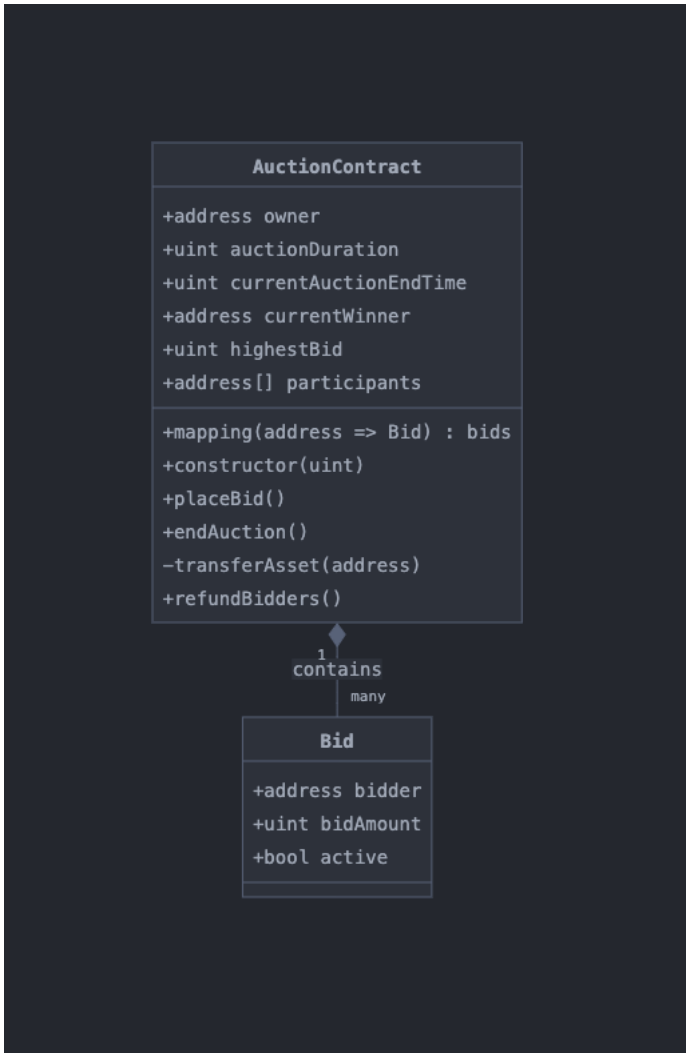


Figure 5:12 Auction smart contract

5.3.3.4 Raffle Smart Contract

The Raffle Smart Contract manages the random selection process where one ROSCA participant receives an asset each cycle through a lottery-based system. The raffle ensures fairness by using a decentralised random number generator (RNG) to select a winner from the pool of eligible participants. The smart contract collects the list of participants, verifies their eligibility (i.e., whether they have made their required contributions for the cycle), and randomly selects one of them as the winner. The contract ensures transparency, fairness, and trustlessness by using an RNG mechanism such as Chainlink VRF to generate a verifiable random outcome. Once the raffle concludes, the contract facilitates the transfer of the asset to the winner, either directly on-chain or by interacting with external oracles for off-chain asset delivery.

Before the raffle can occur, the contract collects all eligible participants who have completed their contributions for the current cycle. Only participants who have fulfilled their obligations for that cycle are allowed to enter the raffle. The contract verifies the contribution status of each participant. Those who have contributed the required amount are automatically added to the raffle pool for that cycle.

To ensure fairness, the contract uses a decentralised RNG service (such as Chainlink VRF) to randomly select a participant from the pool. This ensures the randomness is verifiable and tamper-proof, preventing any manipulation of the raffle results. The contract calls an external RNG service or implements an on-chain RNG function that generates a random number used to select the winning participant.

Once a participant wins the raffle, their eligibility for future raffles is temporarily locked to ensure fairness for other participants. This prevents them from winning consecutive raffles, giving other participants a chance to win future cycles. After the raffle is completed, the contract updates the participant's eligibility status, ensuring that they are excluded from the next raffle cycle.

After the random winner is selected, the contract manages the transfer of the asset to the winner. For physical assets, it may require interacting with an oracle to verify the transfer, while digital assets can be transferred directly on-chain. Once the winner is selected, the contract triggers the asset transfer function. This function either directly transfers digital assets or works with external services to facilitate the delivery of physical goods. The functionality of the smart contract is presented in Figure 5:13.

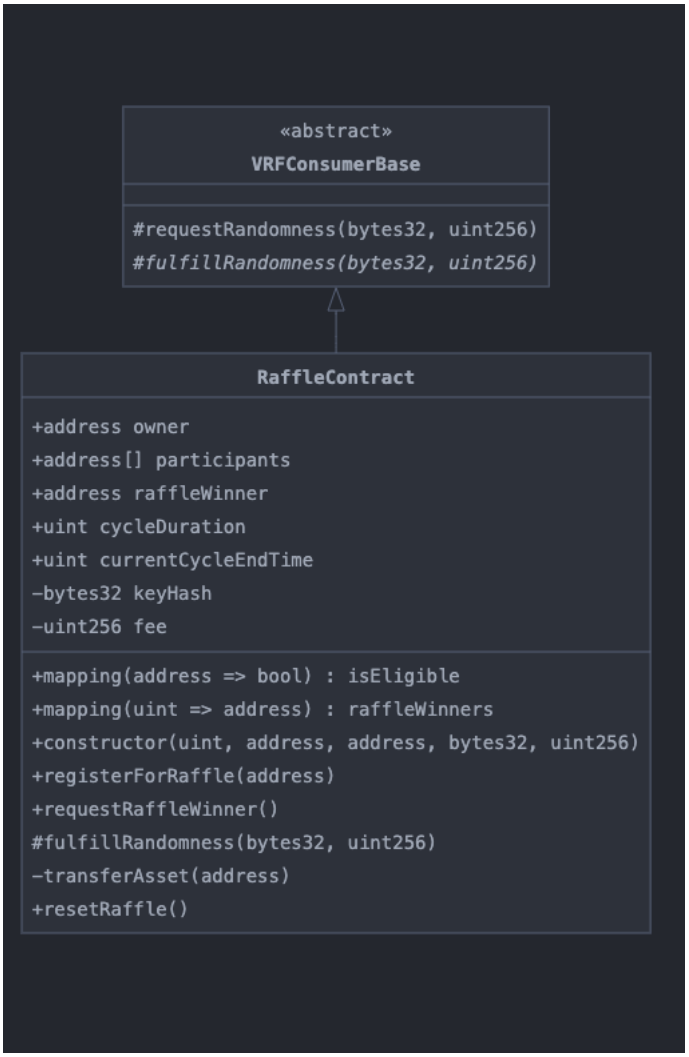


Figure 5:13 Raffle smart contract

5.3.3.5 Settlement Contract

The Settlement Contract handles the final transfer of assets and funds at the end of the ROSCA cycle. This contract ensures that, once a participant has won an auction or raffle, the ownership of the asset is securely transferred from the GO to the participant. It also manages the release of any remaining funds or contributions that need to be returned to the participants at the conclusion of the ROSCA cycle. The contract ensures that all necessary conditions are met before transferring assets, including verifying the status of the winner and confirming that all contributions have been made. For physical assets, it may work with oracles to ensure that off-chain assets (like EVs or heat pumps) are delivered, while digital assets (like energy credits) can be transferred directly on-chain.

The core function of the Settlement Contract is to facilitate the transfer of the asset (e.g., EV, heat pump, flexibility asset) to the participant who won the auction or raffle. The contract ensures that the winner is correctly identified, and the transfer occurs only after all criteria have been met. The contract verifies the winner, either from the auction or raffle, and initiates the transfer. If it's a digital asset, the transfer happens directly on-chain. For physical assets, the contract interacts with an oracle to confirm delivery from the retailer.

At the end of the ROSCA cycle, any remaining funds—such as unallocated contributions or leftover balances—are distributed to participants. This ensures that the cycle concludes with all funds properly accounted for and returned where applicable. The contract calculates any leftover contributions or funds and returns them to the respective participants, either proportionally or based on predefined rules.

The contract finalises the transfer of ownership of assets that were leased to participants during the ROSCA cycle. This means that any asset a participant has been using (such as an EV) is officially transferred to their ownership once the cycle is complete. At the conclusion of the final contribution, the contract ensures that full ownership rights are passed from the GO to the participant, making the asset legally theirs.

Once all settlements have been made and all assets and funds distributed, the contract closes the ROSCA cycle, locking any further transactions or modifications to the group. The contract checks that all outstanding transactions have been settled and then marks the ROSCA as complete. No further activity can take place within the group unless a new cycle is initiated. The functionality of the smart contract is presented in Figure 5:14.

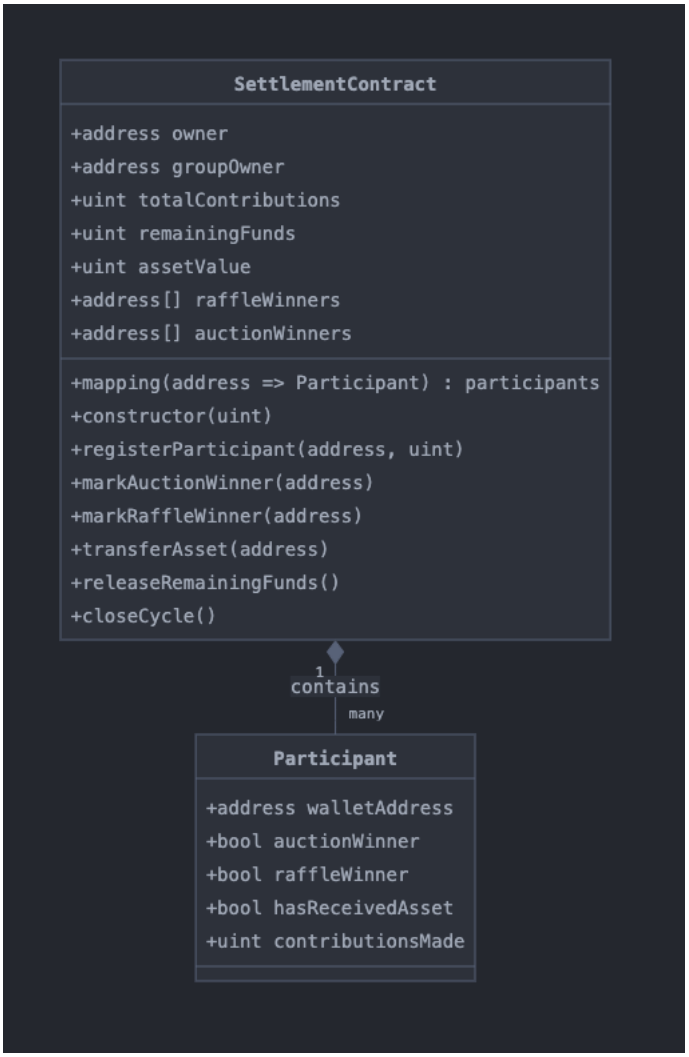


Figure 5:14 Settlement smart contract

5.3.4 Contract workflow

The workflow of the blockchain-based ROSCA system for the energy community begins with the GO Contract, which initiates the entire process by setting the rules, registering participants, and defining the asset type. Once participants are registered and the ROSCA group is full, the system moves into the contribution phase, managed by the Contribution Management Contract. This contract ensures that participants make their monthly contributions as agreed. Each contribution is securely locked in the contract until it is ready to be used for either raffles or auctions.

Next, the Auction Contract and Raffle Contract handle the distribution of assets each cycle. The Auction Contract allows participants to bid additional contributions to receive the asset earlier. Participants submit bids, and at the end of the bidding

period, the highest bid wins. Simultaneously, the Raffle Contract runs a lottery for participants who have contributed to the current cycle. It uses a decentralised random number generator to fairly select a raffle winner.

Once the winners from both the auction and the raffle are determined, the Settlement Contract is triggered. The Settlement Contract plays a crucial role in finalising the ROSCA cycle by transferring the asset to the winners and ensuring all conditions are met before the transfer. For digital assets, the transfer happens directly on-chain, while for physical assets, the contract interacts with oracles to verify the off-chain transfer. Additionally, any remaining funds (e.g., unallocated contributions) are distributed back to the participants as refunds, ensuring fairness.

The Settlement Contract ensures that all participants either receive the asset they contributed towards or are refunded their contributions at the end of the cycle. Once all assets are distributed, and funds are settled, the GO Contract initiates the next ROSCA cycle or closes the group if the process is complete. The architecture and coordination of smart contracts are presented in Figure 5:15.

Figure 5:15 Smart contract architecture flow

5.4 Test-case and results

5.4.1 Test Case

To evaluate the feasibility and effectiveness of the proposed blockchain-based ROSCA system in financing flexibility assets within a residential energy community in the UK, a detailed test case is presented. This case study illustrates the operation of

the system within a specific context, encompassing the community's electrical network specifications, the requirements for flexibility assets, and the interactions between participants and the smart contracts over the ROSCA cycles.

The test case is set in a suburban residential community located on the outskirts of London, comprising 48 households. These households have formed an energy cooperative with the objective of promoting sustainability, reducing carbon emissions, and enhancing energy independence in alignment with the UK's Net Zero 2050 goals. The community is motivated to invest in flexibility assets, such as EVs, to achieve these objectives.

The community is connected to the local distribution network operated by UKPN. The electrical infrastructure comprises an 11 kV medium-voltage distribution network, stepped down to 400/230 V for residential use. This conversion is handled at a local substation equipped with a 1 MVA transformer, which supplies power to the community. The community's peak demand is approximately 500 kW during evening hours, largely driven by residential usage patterns such as lighting, heating, appliances, and entertainment systems.

Figure 5:16 illustrates the network configuration, including the connection to the NGED distribution network, the transformer rating, and voltage levels supplied to the residential area. This setup forms the basis of the test case scenario. The assumed demand levels align with typical values observed at LV substations across NGED's network.

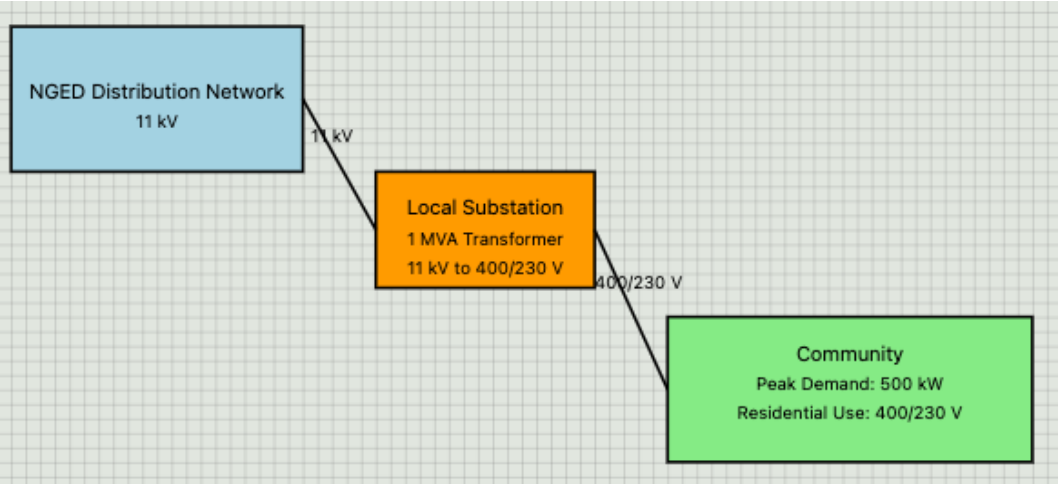


Figure 5:16 Test network

Currently, the community has limited rooftop solar installations contributing about 50 kW during peak solar hours. However, with the anticipated increase in EV charging and the electrification of heating through heat pumps, the transformer and distribution lines are nearing their capacity limits. This situation necessitates the implementation of demand-side flexibility measures to manage the growing energy demands without overloading the existing infrastructure.

To manage the escalating energy demands and optimise the utilisation of renewable energy, the community identifies the need to invest in flexibility assets. Introducing EVs not only reduces transportation emissions but also provides the potential for vehicle-to-grid (V2G) services. EVs can supply energy back to the grid during peak demand periods, offering flexibility.

The community aims to achieve a flexibility capacity of at least 480 kW, equating to 10 kW per household. This level of flexibility would significantly alleviate stress on the local transformer and distribution lines, potentially delaying or eliminating the need for costly infrastructure upgrades. It also enables participation in demand response programs offered by the grid operator, contributing to grid stability and efficiency.

Recognising the high upfront costs associated with these flexibility assets, which in this case are EVs, is averaging £24,000 per unit, the community decides to implement a blockchain-based ROSCA system. This financial arrangement allows members to collectively save and distribute funds to acquire these assets over time, leveraging mutual financial support and trust within the community.

5.4.1.1 Group Formation and Settings

An EV dealership partners with the community, acting as the GO to facilitate the ROSCA. The GO creates a smart contract on a blockchain platform that outlines the group's terms and conditions. The asset of focus is the EV priced at £24,000 each (derived from the cost of a Nissan Leaf). The group consists of 48 households, matching the total number of households in the community. The ROSCA cycle is set to span 48 months, with each household contributing £500 per month.

Two EVs are to be distributed each month: one via a raffle and one via an auction. This structure is designed to ensure that all participants eventually receive an EV while providing opportunities for members to receive their EVs earlier through the auction mechanism. The smart contract locks in these settings, and participants confirm their acceptance, thereby initiating the ROSCA. This has been presented in Figure 5:17.

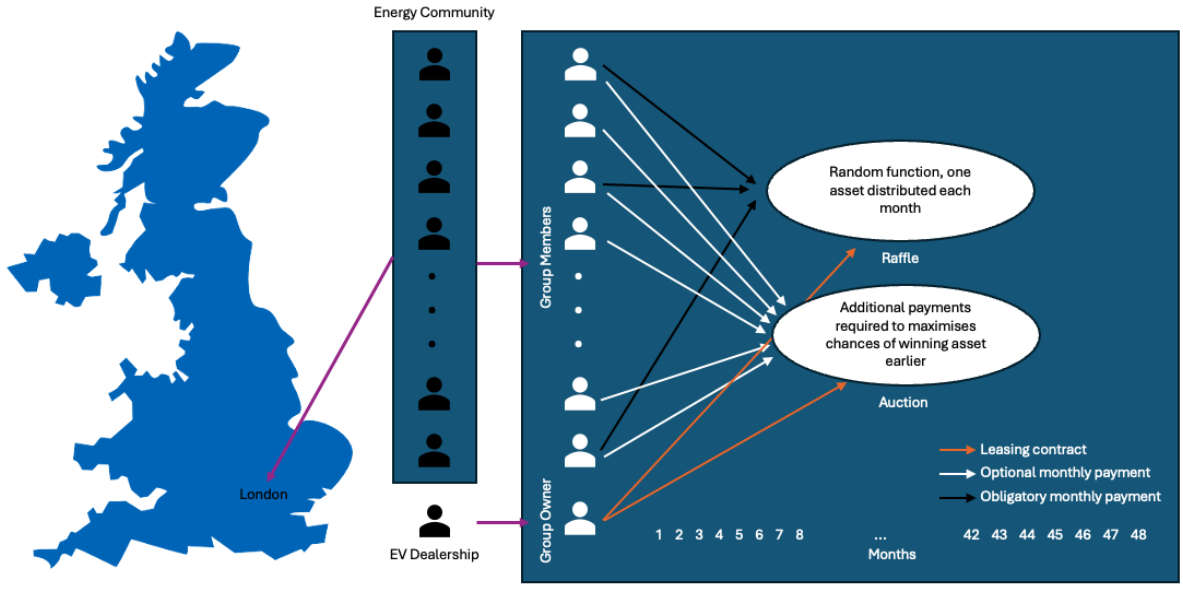


Figure 5:17 ROSCA implementation

5.4.1.2 Operation of the ROSCA System

In the first month, all 48 households make their initial contribution of £500, resulting in total regular contributions of £24,000. This amount is sufficient to cover the cost of one EV. However, to distribute two EVs per month as planned, the system relies on the auction mechanism to generate additional funds.

Participants interested in receiving an EV earlier than the scheduled rotation can participate in a sealed-bid auction. In this month, Household A bids an additional five instalments, amounting to £2,500 (£500 per instalment). Household B bids an additional four instalments, totalling £2,000. These bids are submitted confidentially to the smart contract, which securely records and compares them without revealing the details to other participants.

Household A wins the auction by offering the highest bid. They pay a total of £3,000 for the month, which includes their regular monthly contribution of £500 and their bid amount of £2,500. This upfront payment accelerates their acquisition of the EV and reduces their remaining contribution period by five months, as they have effectively prepaid those instalments. The smart contract updates their contribution schedule accordingly.

The raffle provides an equal opportunity for all contributing households to receive an EV, regardless of their financial capacity to bid in the auction. Excluding Household A, who has already received an EV through the auction, the remaining 47 households are eligible for the raffle. The smart contract utilises a decentralised random number generator, which in this case is Chainlink VRF, to ensure fairness and transparency in the selection process.

In this instance, Household D is randomly selected as the raffle winner. Household C pays only the regular monthly contribution of £500. The smart contract records the outcome and updates the status of Household C, ensuring they are excluded from future raffles to give others a fair chance.

At the end of Month 1, two EVs are distributed: one to Household A, the auction winner, and one to Household D, the raffle winner. The smart contract facilitates the transfer of ownership by interacting with oracles that verify the delivery of the EVs from the dealership to the households. The participants receive the EVs under a leasing arrangement until they complete their full contribution commitments, at which point full ownership is transferred. This arrangement ensures that participants continue to meet their obligations while enjoying the benefits of the asset. This has been showcased in Figure 5:18.

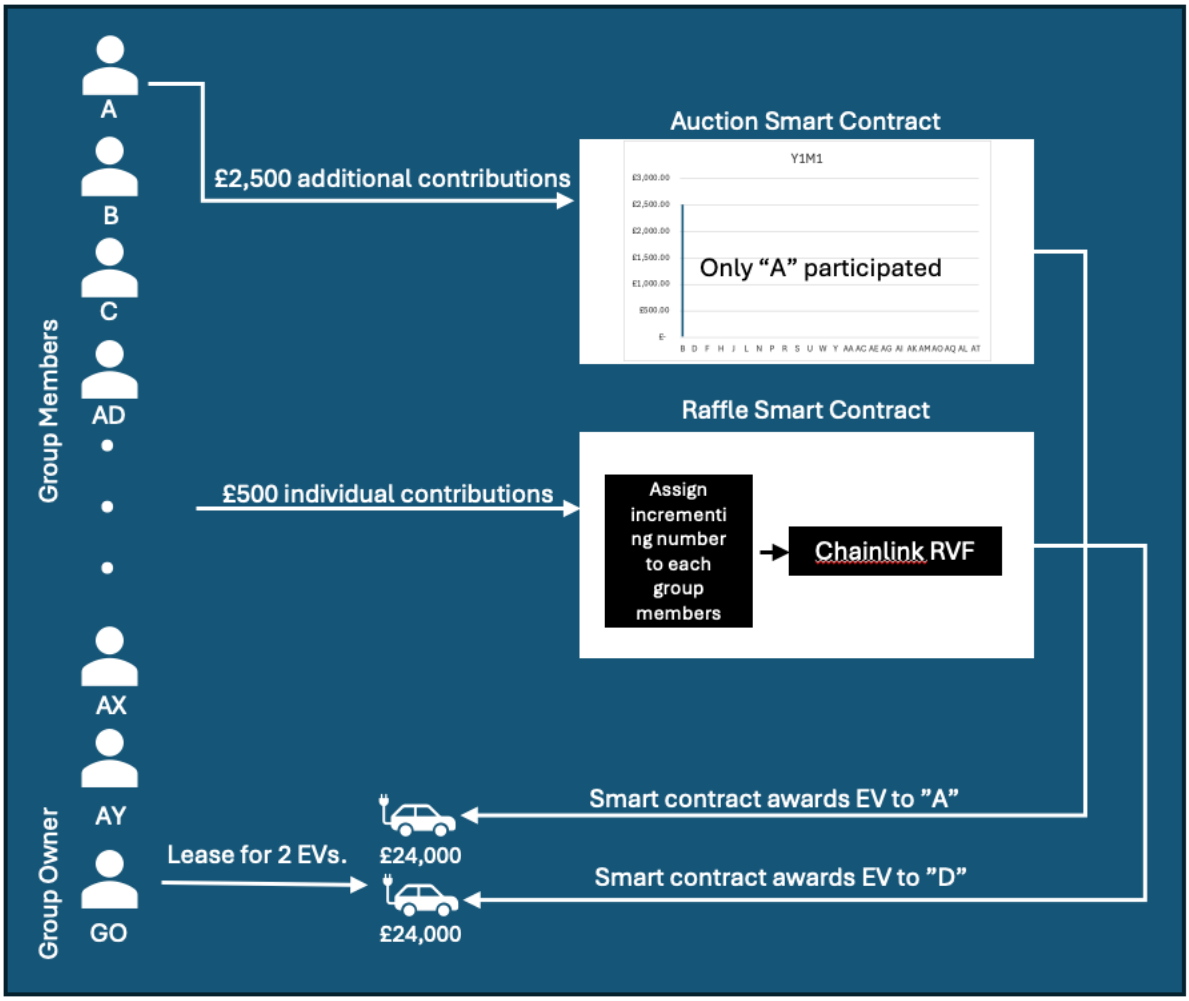


Figure 5:18 Y1M1 ROSCA Operation

As expressed in Figure 5:19, the total funds collected in Month 1 are £24,000 from regular contributions and £2,500 from Household A's bid, totalling £26,500. The cost of two EVs is £48,000. The deficit of £21,500 is managed by the GO, as the leaser. In further months, the extra funds from bids accumulate over time, creating a financial buffer that ensures the sustainability of the ROSCA system. The smart contract keeps track of the financial status, ensuring transparency and accountability.

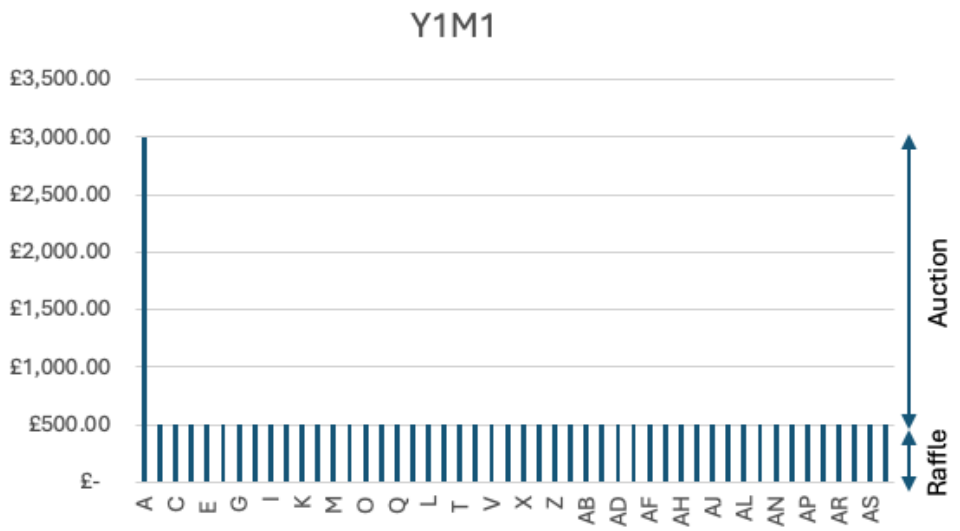


Figure 5:19 Y1M1 Operations

In the second month, Household A does not make a regular contribution, as they have prepaid five instalments through their auction bid. The remaining 47 households contribute £500 each, totalling £23,500. The funds collected are used towards the purchase of two more EVs.

In Month 2, Household C bids an additional four instalments (£2,000), and Household B bids an additional two instalments (£1,000). Household C wins the auction with the highest bid. They pay a total of £2,500 for the month (£500 regular contribution + £2,000 bid). Like Household A, Household C reduces their remaining contribution period by four months. The smart contract adjusts their payment schedule and locks in their commitment.

Excluding previous recipients, 45 households are eligible for the raffle. The smart contract performs the random selection, and Household F is selected as the raffle winner. Household F pays only the regular contribution of £500. The smart contract updates their status to reflect the receipt of the EV and their exclusion from future raffles.

Two EVs are distributed in Month 2: one to Household C, the auction winner, and one to Household F, the raffle winner. The transfer of assets follows the same process as in Month 1, with the smart contract ensuring proper execution and

record-keeping. The oracles confirm the delivery of the EVs, and the leasing arrangements are established. This has been showcased in Figure 5:20.

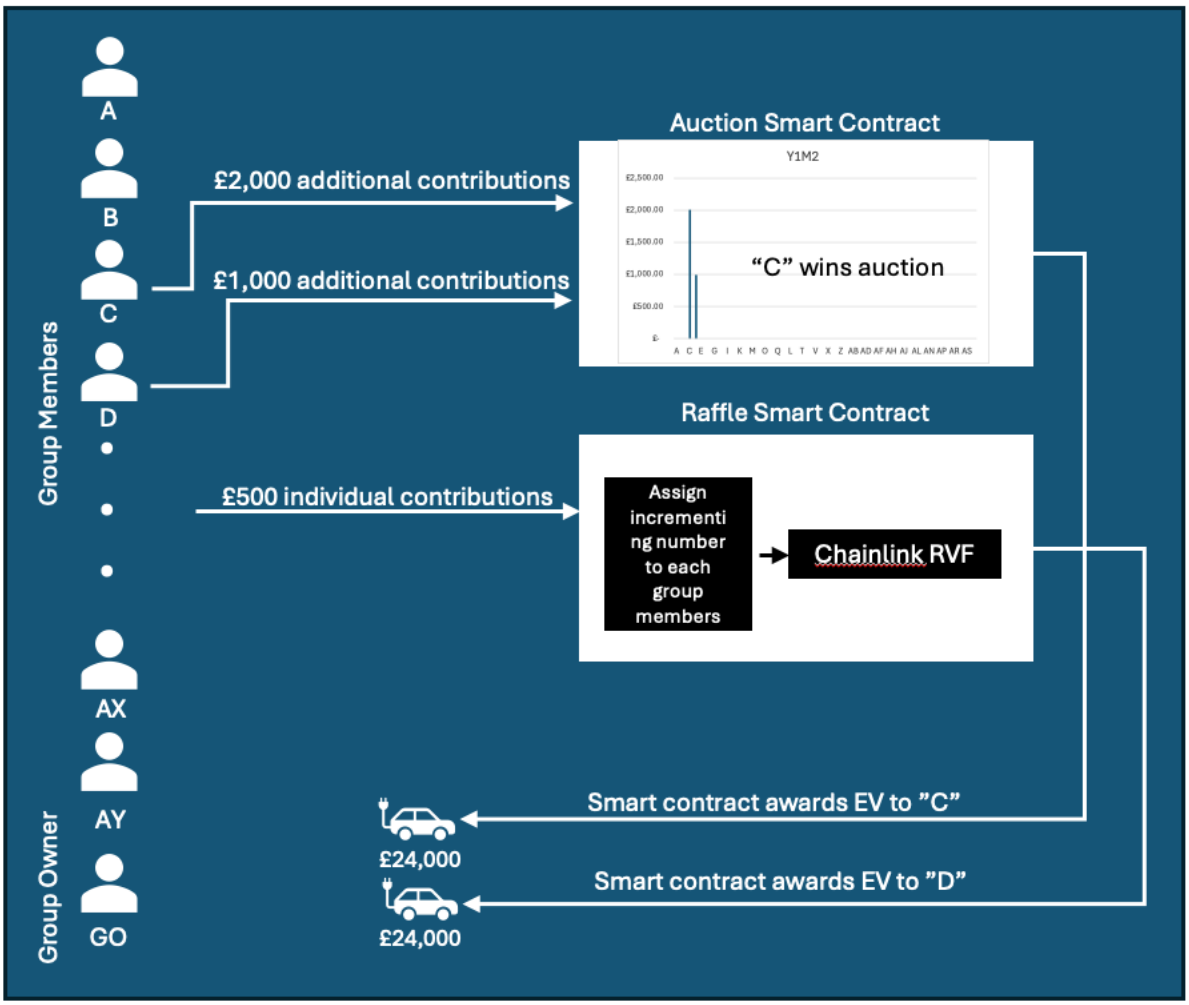


Figure 5:20 Y1M2 ROSCA Operation

As presented in Figure 5:21, the total funds collected in Month 2 are £23,500 from regular contributions and £2,000 from Household D's bid, totalling £25,500. The smart contract maintains an updated ledger of all transactions, providing participants with access to financial statements.

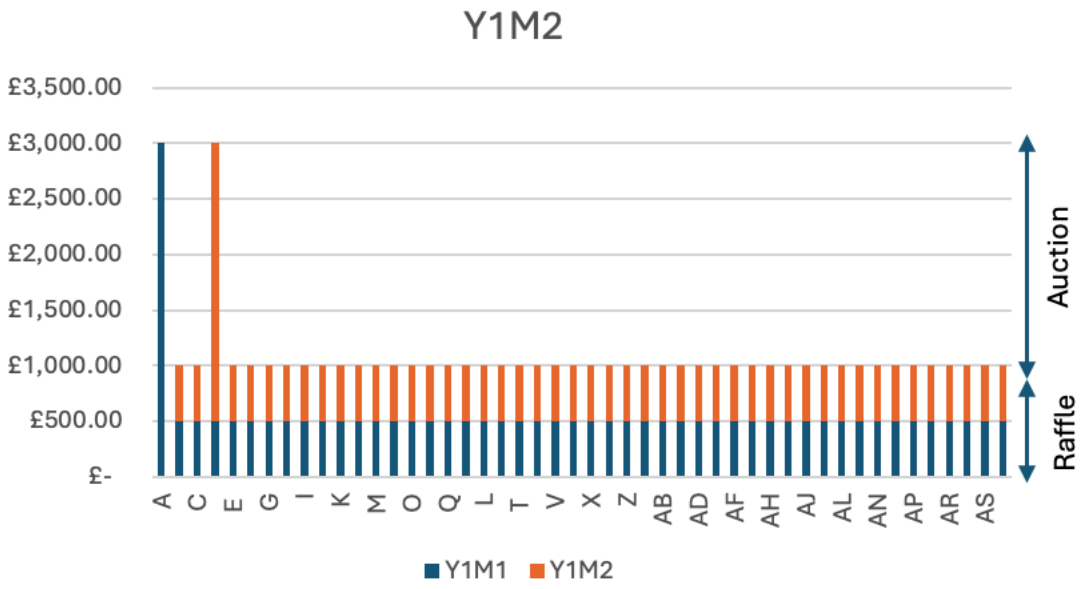


Figure 5:21 Y1M2 operation

Over the following months, the process repeats. Households continue to make regular contributions unless they have prepaid instalments through auction bids. As more households receive EVs and some complete their contribution obligations earlier due to prepayments, the total monthly contributions adjust accordingly. The following are the results at the end of the 48 months. All the cars have been distributed by Y3, with the following months all households finalising the remaining payments. Each household will have the ownership transferred to the by the GO at the end of Y4. The total contributions over the 48 months period for each of the participants are presented in Figure 5:22.

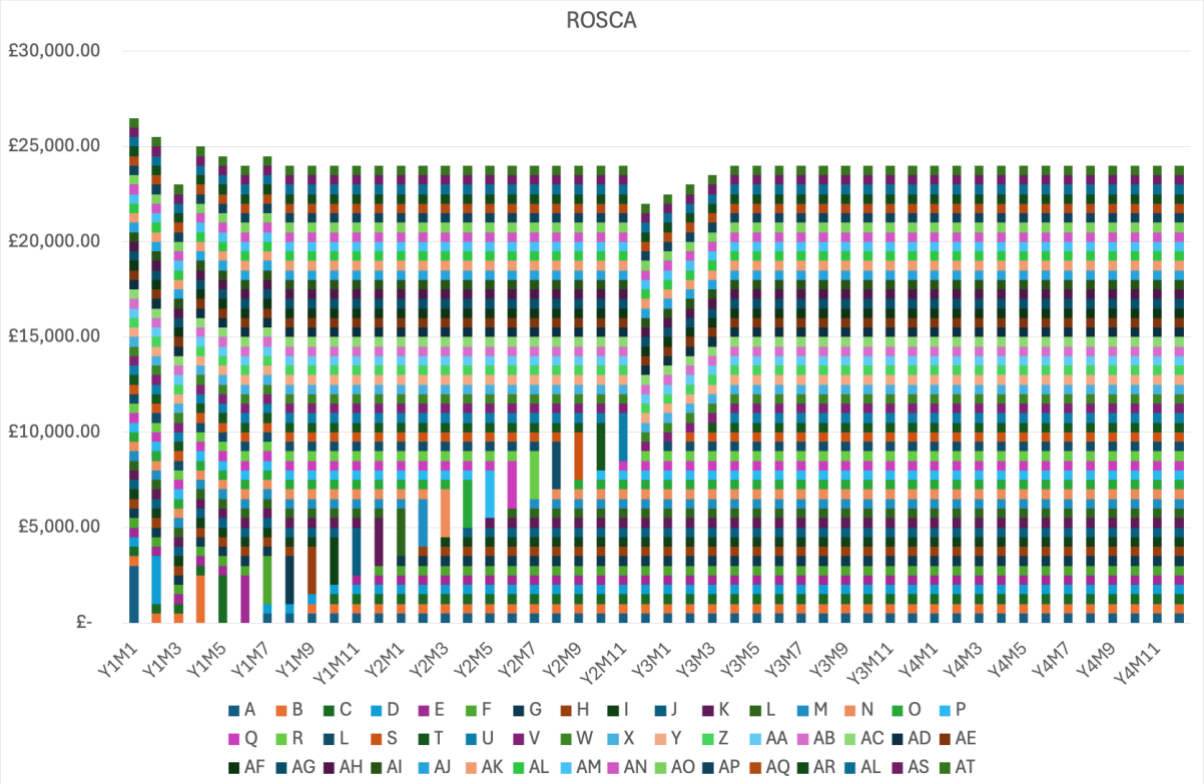


Figure 5:22 ROSCA end period results

The smart contracts manage all transactions, contributions, bids, and asset distributions automatically, ensuring transparency, security, and adherence to the agreed-upon rules. Participants can monitor the status of the ROSCA, their contributions, and their remaining obligations through a user-friendly interface connected to the blockchain platform.

5.4.2 Analysis of Results

By the end of Y2, all 48 households have acquired EVs. Assuming each EV can provide up to 10 kW of flexibility through V2G services, the community achieves a combined flexibility capacity of 480 kW. This capacity allows the community to effectively manage peak loads, participate in demand response programs, and integrate more renewable energy sources without overloading the local distribution network. The community can offer significant flexibility services to the grid operator, potentially generating additional revenue and contributing to grid stability. In the context of UK, if the ROSCA would begin operating at the end of 2024, the energy community will have acquired the maximum capacity of flexibility. This is then showcased in Figure 5:23.

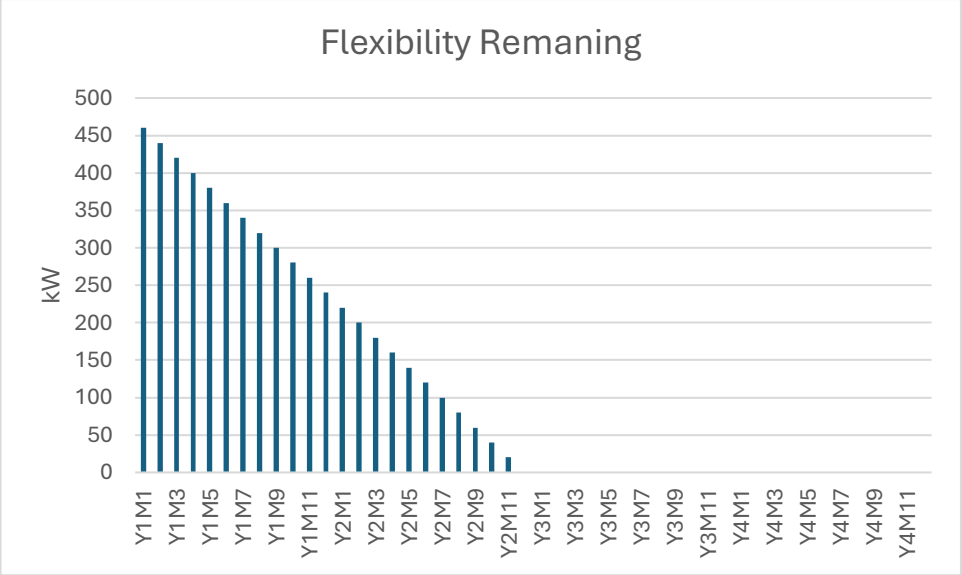


Figure 5:23 Flexibility left to be fulfilled over time

The ROSCA system enables households to acquire expensive assets without relying on traditional financing mechanisms, which often involve interest payments and stringent credit requirements. By pooling resources and supporting each other, the community members reduce individual financial burdens. This is mostly efficient towards the Net Zero transition on fuel poor communities, that might not benefit from a credit score or additional money to allow them to purchase the assets from the first month. The auction mechanism efficiently allocates assets to those willing and able to receive them earlier by paying additional instalments, while the raffle ensures that all participants have an equal chance of receiving the asset sooner, maintaining fairness and inclusivity.

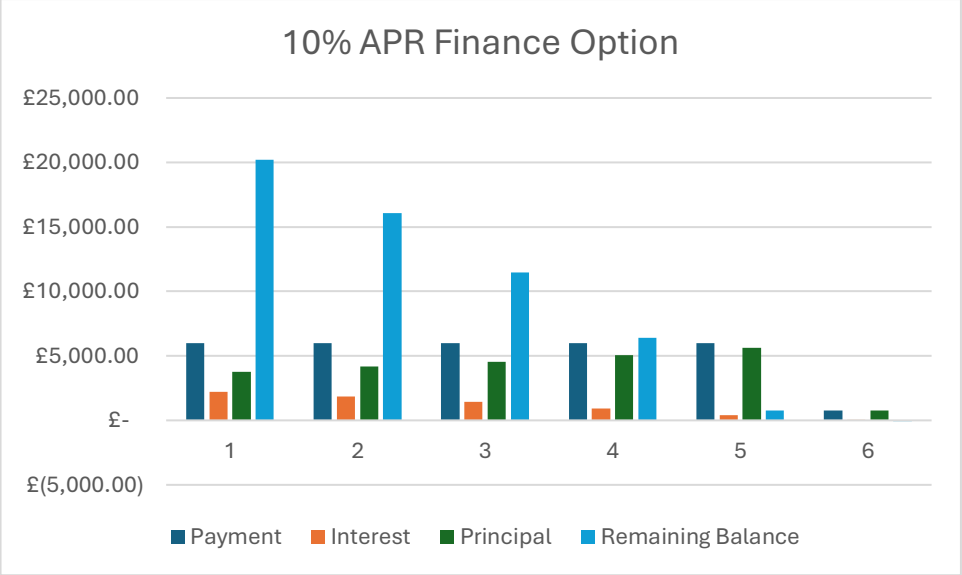


Figure 5:24 10% APR finance option

As presented in Figure 5:24, when compared to traditional financing methods where households individually purchase EVs through loans at a 10% annual percentage rate, the ROSCA system presents significant financial advantages. In the conventional loan scenario, a household pays £6,000 annually over six years, amounting to a total of approximately £30,813, including £6,813 in interest payments. This means the household pays more than the EV's purchase price and gains full ownership only after completing all loan payments, delaying asset ownership. In contrast, the ROSCA system eliminates interest charges, allowing households to pay only the EV's purchase price of £24,000, resulting in immediate savings of around £6,813 per household. When extended to the entire community of 48 households, the collective savings are substantial—over £327,000 in interest payments alone. This not only reduces the financial burden on individual households but also accelerates ownership and frees up funds that can be reinvested into additional sustainability initiatives within the community. Thus, the ROSCA model proves to be a more cost-effective and efficient financing option, aligning financial incentives with the community's environmental goals.

From the GO perspective—as the EV dealership facilitating asset distribution—two distinct cashflow dynamics emerge based on the underlying mechanism used: raffle and auction. In the raffle model, the GO operates on a cash-neutral basis. Each month, all 48 participating households contribute £500, resulting in a total of

£24,000—exactly the amount required to fund one EV. The EV is then allocated through a randomised draw. This structure ensures that the GO does not need to borrow or front any capital, as the monthly inflow fully covers the asset outflow. However, minor temporary imbalances may occur due to variable contribution patterns and smart contract rounding, with a small, accumulated deficit (approximately £7,000) appearing mid-way through the cycle. This is fully resolved in the final months as all participants complete their contributions, restoring the cash position to zero by Month 48. Figure 5:25 presents the accumulative raffle cashflow for the 48 months period.

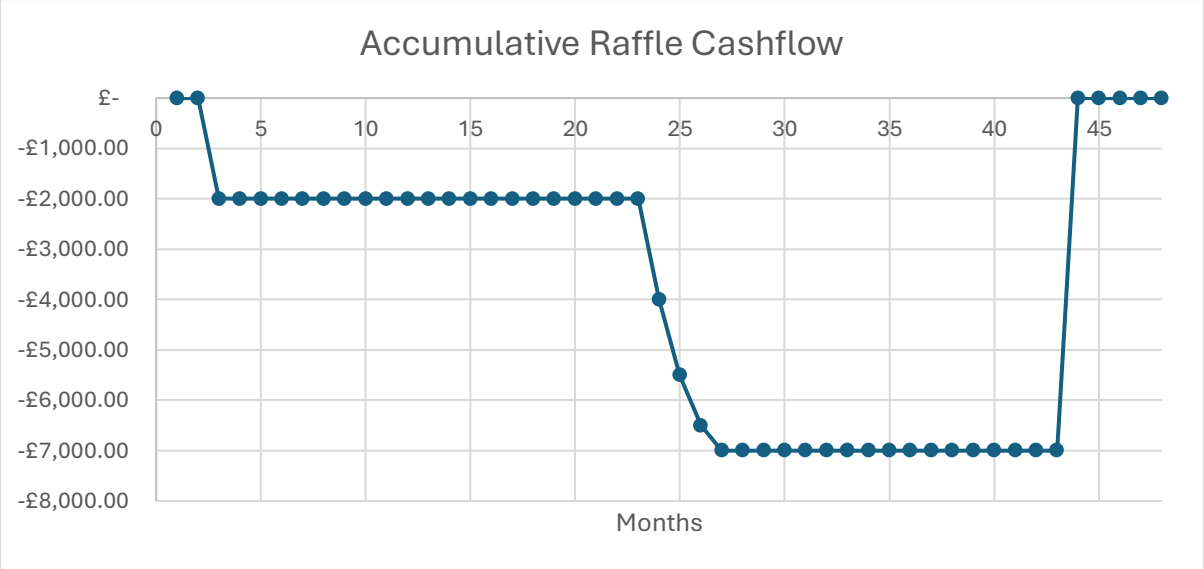


Figure 5:25 Accumulative Raffle Cashflow

Conversely, the auction model introduces an accelerated distribution strategy in which two EVs are delivered per month. While participants may bid to receive an EV earlier, thereby prepaying a portion of their contributions upfront, the overall income still lags behind the capital demand for vehicle procurement in the early phase. For example, in Month 1, although all 48 participants contribute their standard £500 (£24,000 total), only one EV is affordable. The second EV is enabled through the auction mechanism, where one participant (e.g., Household A) offers an additional £2,500 in advance to secure early access. This approach results in a substantial cashflow deficit for the GO during the first two-thirds of the ROSCA timeline—reaching over £110,000 in deficit by Month 5. This deficit remains static for most of the project duration, as contributions continue at a predictable pace while EV

distribution is front-loaded. However, in the final months, when all EVs have already been distributed and the only remaining activity is repayment, the GO's financial position gradually improves until the cumulative deficit is eliminated by Month 48. These cashflow patterns illustrate the strategic trade-off available to the GO. The raffle model offers financial stability and zero exposure but results in slower asset turnover. In contrast, the auction model accelerates asset delivery and enhances participant engagement but requires the GO to accept a temporary negative balance. Importantly, the system's smart contracts and transparent accounting allow the GO to make this decision proactively, choosing whether to absorb the financing burden in exchange for a more dynamic and appealing participation model. Figure 5:26 presents the accumulative raffle cashflow for the 48 months period.

Figure 5:26 Accumulative Auction Cashflow

The smart contracts ensure that all financial transactions are transparent and tamper-proof. Participants can verify their contributions, bids, and the allocation of assets at any time. The elimination of intermediaries reduces administrative costs and enhances efficiency. Implementing the ROSCA on a blockchain platform significantly enhances trust among participants. The use of smart contracts ensures that all transactions and processes are executed automatically and according to the predefined rules, eliminating the risk of human error or manipulation. The immutable

nature of blockchain records provides a transparent and tamper-proof ledger of all contributions, bids, asset distributions, and settlements. Participants can verify the integrity of the system at any time, fostering confidence in the process. The decentralised nature of the blockchain reduces reliance on a central authority, aligning with the cooperative principles of the community. The smart contracts' code is open for audit, allowing participants to understand and trust the mechanisms governing the ROSCA.

The ROSCA system fosters a sense of community and mutual support among members. Participants benefit from flexible options to receive assets earlier through auctions if they are willing and able to contribute more upfront, or by chance through raffles, accommodating different financial capacities and preferences. The excitement and anticipation associated with the possibility of receiving an EV earlier enhance engagement and commitment to the program.

5.5 Conclusion

The detailed test case demonstrates that the proposed blockchain-based ROSCA system effectively facilitates the acquisition of flexibility assets within a residential energy community in the UK. By combining cooperative finance with advanced blockchain technology, the community achieves its sustainability goals, enhances energy independence, and contributes to grid stability and decarbonisation efforts.

The ROSCA model proves to be an inclusive and efficient mechanism, enabling households to overcome financial barriers associated with high upfront costs of flexibility assets. The integration of smart contracts ensures transparency, security, and automation, fostering trust and active participation among community members.

Furthermore, the successful deployment of flexibility assets like EVs significantly enhances the community's ability to manage its energy demands, integrate renewable energy sources, and support the broader electricity network through demand response and flexibility services. This model serves as a viable blueprint for other communities aiming to transition towards sustainable and decentralised energy systems in alignment with national and global climate objectives.

By implementing this financial and technological approach, the community not only advances its own sustainability objectives but also contributes to the collective effort to mitigate climate change. The test case underscores the potential of blockchain-based ROSCA systems to drive social and environmental progress, highlighting their applicability in diverse contexts where cooperative action and technological innovation converge to address pressing challenges.

6 Conclusion and future work

6.1 Conclusion

This thesis has explored the use of blockchain-based architectures to support decentralised flexibility services in electricity distribution networks. Three core technical contributions were presented: a distributed ledger framework for local flexibility markets, a ZKP mechanism for privacy-preserving settlement, and a blockchain-enabled ROSCA model for financing flexibility assets. Collectively, these contributions demonstrate the potential for DLTs to play a foundational role in shaping fair, transparent, and resilient energy systems. Each technical chapter has achieved its objectives but also revealed key limitations that must be critically examined.

The first technical chapter introduced a full stack blockchain framework for a local flexibility market. While the model proved functional in simulating market processes such as bid registration, market clearing, and settlement, its evaluation remained largely bounded within a test environment. The results confirmed that both Ethereum and Hedera Hashgraph could support end-to-end market functionality, but also exposed challenges around throughput, finality speed, and integration with real-world infrastructure. Additionally, the framework assumed homogeneous flexibility products and full digital readiness, which may not reflect the current operational reality of DNOs. Thus, although promising, the platform needs further testing under more diverse scenarios—including mixed asset portfolios, varied price signals, and multi-platform integration.

In particular, the experiments highlighted that while Hedera offers greater throughput and lower finality times, it lacks the open developer ecosystem and smart contract flexibility of Ethereum. These platform trade-offs suggest that the choice of blockchain must be aligned with specific deployment goals—whether that be performance, cost-efficiency, or community-driven innovation. Future work may benefit from hybridising blockchain components to leverage the strengths of both public and permissioned infrastructures.

The second technical chapter advanced the state of the art by implementing a ZKP-based settlement method. This mechanism allows for the validation of flexibility delivery without disclosing users' private electricity consumption data. The model, tested in the MINA blockchain ecosystem, functioned as intended with standard baselining methodologies. However, it has limitations in terms of computational complexity, and its evaluation was restricted to a narrow set of baselining strategies. Moreover, the practical feasibility of using ZKPs at scale—across a large number of DERs and aggregators—remains unverified. While the privacy benefits are clear, further scrutiny is needed to assess user trust, system resilience, and long-term maintainability.

The implementation also relied on simplified assumptions about the data structure and submission process for baselines. In practice, flexibility service providers operate with varying reporting formats, non-linear response profiles, and multi-asset aggregations (historically in units). Translating these real-world complexities into efficient ZKP circuits remains a key technical hurdle. Additionally, the cost and time associated with generating proofs, especially for resource-constrained participants, could introduce adoption barriers if not carefully optimised.

The final technical chapter proposed a decentralised ROSCA framework to democratise access to flexibility assets, especially within community energy contexts. The model enables transparent, rule-based asset allocation using smart contracts to manage both raffle and auction mechanisms. Cashflow modelling provided a realistic view of the GO’s exposure and the system’s sustainability. Still, this chapter too presented several limitations. Behavioural assumptions (such as rational participation and low default risk) need further testing, and the financial

model has yet to be evaluated across a wider socioeconomic spectrum. Furthermore, regulatory and legal constraints around decentralised finance for energy assets were not explored in depth.

For example, the legality of peer-to-peer financial instruments and pooled ownership models varies considerably across jurisdictions. Smart contract execution in financial contexts may trigger licensing requirements or consumer protection obligations, particularly in the UK or EU under emerging crypto-regulation frameworks. These considerations must be explored before ROSCAs can be confidently deployed as mainstream financial mechanisms in energy systems.

Across all chapters, the thesis has made a meaningful contribution by aligning technical architecture with the broader goals of system decarbonisation and decentralised governance.

6.2 Future Work

The future direction of this research is twofold: first, to overcome the limitations identified in each technical chapter, and second, to extend the scope of the work into new operational, regulatory, and societal contexts.

For the local flexibility market platform, the most pressing next step is to test the system in live operational environments, ideally in collaboration with a DNO. Doing so would uncover technical and governance challenges that are difficult to simulate, such as latency under real-world traffic, smart meter interoperability, and digital identity management for new participants. Additionally, the platform should be extended to support more granular market segmentation, including sub-half-hourly products and co-optimisation across network constraints and carbon signals. Developing interfaces with national markets (e.g., for ancillary services) would also enable revenue stacking and provide stronger business cases for DER participation.

Further development should also include mechanisms for dispute resolution, fallback procedures during network downtime, and automated compliance logging to satisfy data retention and audit requirements. This could involve embedding regulatory logic

within smart contracts or designing off-chain governance interfaces that allow DNOs to intervene, when necessary, without compromising decentralisation goals. In this way, the framework can strike a balance between automation and oversight.

For the ZKP-based settlement model, future research should evaluate a broader set of baselining methodologies, such as dynamic reference profiles or peer-comparison benchmarks. Moreover, the performance of the ZKP algorithm must be benchmarked under scale, ideally in conditions where multiple aggregators are submitting proofs simultaneously. The use of alternative privacy-preserving technologies, such as homomorphic encryption or secure multi-party computation, should also be investigated, particularly where ZKPs may prove computationally intensive. Additional focus should be placed on regulatory acceptance, especially concerning how DNOs and regulators interpret zero-knowledge attestations in formal compliance processes.

The ROSCA model offers substantial scope for further work. One avenue involves testing the model under behavioural stress conditions, such as late payments, dropout risk, or coordinated manipulation of the auction system. Additionally, the system could be adapted for use in developing regions, where traditional banking systems are absent and decentralised finance tools may offer high impact. Future studies should also consider integrating reputation-based mechanisms or smart credit scoring to increase participation and reduce moral hazard. More broadly, the model could be expanded to support other forms of clean energy infrastructure, such as home batteries or community solar arrays.

At a systemic level, future research must explore regulatory, institutional, and interoperability pathways for scaling blockchain-based energy systems. This includes investigating how smart contracts can be designed to comply with existing legal frameworks, how governance models (e.g. decentralised autonomous organisation-like structures) can be introduced responsibly, and how open standards can facilitate multi-party data exchange. Research should also consider social adoption factors, including how different communities understand, trust, and engage with decentralised systems—especially those involving financial risk and shared infrastructure. This includes attention to user experience design, language

accessibility, and the framing of incentives. For example, participants in community energy projects may require clear visual interfaces to understand their contributions, allocations, and expected returns over time. Trust in the system is unlikely to stem solely from technical assurance: social validation, peer recommendations, and trusted intermediaries may play just as important a role. Incorporating these aspects will be key in scaling up from prototypes to meaningful adoption.

Finally, while this thesis has demonstrated the conceptual and technical feasibility of decentralised flexibility systems, their realisation depends on pilot deployment and field validation. Collaborations with utilities, aggregators, technology providers, and regulators can bring these models into practice. Field trials will offer insight into the viability, benefits, and unintended consequences of blockchain-based flexibility systems—and will ultimately determine whether they can scale into critical infrastructure for a Net Zero future.

7 REFERENCES

Citizen Advice (2019) How Energy Disputes Are Resolved.

Agency, I.R.E. (2019) Innovation Landscape for a Renewable-Powered Future: Solutions to Integrate Variable Renewables. IRENA, Abu Dhabi.

Aitzhan, N.Z. and Svetinovic, D. (2018) 'Security and Privacy in Decentralized Energy Trading Through Multi-Signatures, Blockchain and Anonymous Messaging Streams', IEEE Transactions on Dependable and Secure Computing, 15, pp. 840–852.

Andoni, M. et al. (2019) 'Blockchain technology in the energy sector: A systematic review of challenges and opportunities', Renewable and Sustainable Energy Reviews, 100, pp. 143–174.

Ardener, S. (1964) 'The comparative study of rotating credit associations', Journal of the Royal Anthropological Institute of Great Britain and Ireland, 94, pp. 201–229.

Ardener, S. and Burman, S. (1995) Money-go-rounds: The importance of rotating savings and credit associations for women. Berg Publishers.

Energy Networks Association (2020) Flexibility Platforms in the UK Energy System.

Energy Networks Association (2019) Our Six Steps for Delivering Flexibility Services.

PoA Authority LLC (2024) PoA.network website. (Accessed: 14 March 2024).

Baird, L. (2022) Method and apparatus for a distributed database within a network. (Accessed: 22 May 2024).

Baird, L., Harmon, M. and Madsen, P. (2022) Hedera: A Governing Council & Public Hashgraph Network. (Accessed: 28 February 2024).

Banerjee, M. et al. (2018) 'Blockchain-Based Security Layer for Identification and Isolation of Malicious Things in IoT: A Conceptual Design', in: 2018 27th International Conference on Computer Communication and Networks (ICCCN). Hangzhou, pp. 1–6.

World Bank (2022) Pakistan Floods 2022: Post-Disaster Needs Assessment.

Bano, S. (2022) Consensus in the Age of Blockchains. (Accessed: 19 April 2024).

UK Department for Business, Energy and Industrial Strategy (2018) Smart Metering Implementation Programme.

Ben-Sasson, E. et al. (2014) 'Zerocash: Decentralized anonymous payments from Bitcoin', in: 2014 IEEE Symposium on Security and Privacy. IEEE, pp. 459–474.

Bouman, F.J.A. (1979) ‘Rotating and accumulating savings and credit associations: A development perspective’, World Development, 7, pp. 369–384.

Bracciali, A., Pintore, F. and Sala, M. (2018) ‘WTSC18 Overview’, in: Pre-Proceedings of the Second Workshop on Trusted Smart Contracts. Curaçao, p. 4.

Central Bank of Brazil (2015) Relatório de Inclusão Financeira.

Brooklyn MicroGrid (2022) Brooklyn MicroGrid website. (Accessed: 07 June 2024).

Brown, D., Hall, S. and Davis, M.E. (2019) ‘Prosumers in the post subsidy era: An exploration of new prosumer business models in the UK’, Energy Policy, 135, 110984.

Burgwinkel, D. (2016) ‘Potential of the blockchain technology in energy trading’, in: Blockchain Technology Introduction for Business and IT Managers. De Gruyter.

UK Department for Business, Energy and Industrial Strategy (2020) The Ten Point Plan for a Green Industrial Revolution.

Buterin, V. (2014) A Next-Generation Smart Contract and Decentralized Application Platform. Ethereum White Paper. (Accessed: 21 August 2024).

Buterin, V. and Griffith, V. (2017) Casper the Friendly Finality Gadget. (Accessed: 12 May 2024).

Byres, E.J., Franz, M. and Miller, D. (2014) ‘The Use of Attack Trees in Assessing Vulnerabilities in SCADA Systems’, in: IEEE International Infrastructure Survivability Workshop (IISW), p. 9.

Cachin, C. (2016) Architecture of the Hyperledger Blockchain Fabric. (Accessed: 29 October 2024).

Caramizaru, A. and Uihlein, A. (2020) Energy communities: an overview of energy and social innovation. European Commission Joint Research Centre.

Cardano (2022) Cardano Web Page. (Accessed: 10 July 2024).

Castro, M. and Liskov, B. (2002) ‘Practical Byzantine fault tolerance and proactive recovery’, ACM Transactions on Computer Systems, 20, pp. 398–461.

Energy Systems Catapult (2020) Local Energy Market Platforms: An Overview. Energy Systems Catapult, Birmingham, UK.

Energy Systems Catapult (2019) Innovating to Net Zero. Energy Systems Catapult, Birmingham, UK.

National Cyber Security Centre (2020) Cyber Threats to the UK Energy Sector.

Chia (2022) Chia platform webpage. (Accessed: 05 March 2024).

Christidis, K. and Devetsikiotis, M. (2016a) ‘Blockchains and smart contracts for the Internet of Things’, IEEE Access, 4, pp. 2292–2303.

Christidis, K. and Devetsikiotis, M. (2016b) ‘Blockchains and smart contracts for the Internet of Things’, IEEE Access, 4, pp. 2292–2303.

Committee on Climate Change (2019) Net Zero – The UK’s Contribution to Stopping Global Warming. CCC, London.

Cointelegraph (2018) ‘The world’s first peer-to-peer energy trading platform, SunContract, launched’. (Accessed: 17 April 2024).

Company, D.C. (2019) Privacy Controls for Smart Metering Data.

Constantinides, P., Henfridsson, O. and Parker, G.G. (2018) ‘Introduction—Platforms and Infrastructures in the Digital Age’, Information Systems Research, 29, pp. 381–400.

Pöyry Consulting (2017) Roadmap for Flexibility Services to 2030. London: Pöyry.

European Union Agency for Cybersecurity (2019) Cybersecurity Challenges in the Uptake of Smart Grid Technology.

Department for Business, Energy & Industrial Strategy (2020) BEIS Annual Report. BEIS, London.

Destefanis, G. et al. (2018) ‘Smart contracts vulnerabilities: A call for blockchain software engineering?’, in: International Workshop on Blockchain Oriented Software Engineering (IWBOSE). Campobasso, pp. 19–25.

Dfinity (2022) Dfinity webpage. (Accessed: 22 October 2024).

UK Government Office for Science (2022) Distributed Ledger Technology: beyond blockchain. (Accessed: 04 May 2024).

Drift Marketplace Inc. (2022) Drift Marketplace website. (Accessed: 30 July 2024).

Dütsch, G. and Steinecke, N. (2017) ‘Use cases for blockchain technology in energy and commodity trading’, in: Snapshot of Current Developments of Blockchain in the Energy and Commodity Sector. PwC.

Electron (2022) Electron website. (Accessed: 08 June 2024).

ELEXON (2021) Settlement Calculation and Allocation.

ELEXON (2021) Data Collection and Aggregation for Settlement Purposes.

ELEXON (2020) Data Transfer and Settlement Processes.

ENA Open Networks Project (2021) Open Networks Project Annual Review 2021. (Accessed: 05 February 2024).

Energy Networks Association (ENA) (2021) Annual Report. (Accessed: 12 February 2024).

Energy Networks Association (ENA) (2018) Flexibility Services Explained. (Accessed: 15 February 2024).

Energy Networks Association (ENA) (2022) Distribution System Flexibility Roadmap. (Accessed: 01 February 2024).

Energy UK (2018) Barriers to Entry for Flexibility Providers.

EnerNOC (2011) The Demand Response Baseline.

ENISA (2019) Cybersecurity Challenges in the Uptake of Smart Grid Technology.

National Grid ESO (2021) Future Energy Scenarios 2021.

National Grid ESO (2020) Baseline Methodologies for Demand Side Response.

Ethereum Classic (2022) Ethereum Classic webpage. (Accessed: 03 February 2024).

Ferrag, M.A. et al. (2018) 'Blockchain technologies for the Internet of Things: Research issues and challenges', IEEE Internet of Things Journal, 6(2), pp. 2188–2204.

Filecoin (2022) Filecoin webpage. (Accessed: 09 October 2024).

Ethereum Foundation (2020) ERC20 Token Standard. (Accessed: 18 April 2024).

Energy Web Foundation (2018) The Energy Web Chain: Accelerating the Energy Transition with an Open-Source, Decentralized Blockchain Platform. (Accessed: 27 May 2024).

Stellar Development Foundation (2022) Stellar website. (Accessed: 16 August 2024).

Frei, C. (2018) The Developing Role of Blockchain, White paper v1.0. World Economic Forum. (Accessed: 10 November 2024).

Smart Energy GB (2021) Understanding Smart Meters.

Geertz, C. (1962) 'The rotating credit association: A "middle rung" in development', Economic Development and Cultural Change, 10(3), pp. 241–263.

Gilad, Y. et al. (2017) 'Algorand: Scaling Byzantine agreements for cryptocurrencies', in: Proceedings of the 26th Symposium on Operating Systems Principles - SOSP '17, Shanghai, China, pp. 51–68.

Goranovic, A. et al. (2017) 'Blockchain applications in microgrids: An overview of current projects and concepts', in: 2017 IEEE International Conference on Industrial Technology (ICIT). IEEE, pp. 1378–1383.

Grid+ (2017) Welcome to the Future of Energy – White Paper v2.0. (Accessed: 06 March 2024).

Guerrero, J., Chapman, A.C. and Verbič, G. (2021a) 'Decentralized P2P energy trading under network constraints in a low-voltage network', IEEE Transactions on Smart Grid, 10(5), pp. 5163–5173.

Guerrero, J., Chapman, A.C. and Verbič, G. (2021b) 'Privacy-preserving techniques for smart grids: A survey', IEEE Access, 9, pp. 28771–28788.

He, H. and Yan, J. (2016) 'Cyber-physical attacks and defences in the smart grid: A survey', IET Cyber-Physical Systems: Theory and Applications, 1(1), pp. 13–27.

He, Q. et al. (2018) 'On the consensus mechanisms of blockchain/DLT for Internet of Things', in: 2018 IEEE 13th International Symposium on Industrial Embedded Systems (SIES), Graz, Austria, pp. 1–10.

He, Y., Mendis, G.J. and Wei, J. (2017) 'Real-time detection of false data injection attacks in smart grid: A deep learning-based intelligent mechanism', IEEE Transactions on Smart Grid, 8(5), pp. 2505–2516.

Hedera Hashgraph (2022) Hedera Hashgraph website. (Accessed: 02 May 2024).

Henry, R., Herzberg, A. and Kate, A. (2018) 'Blockchain access privacy: Challenges and directions', IEEE Security & Privacy, 16(4), pp. 38–45.

Hewa, T., Yuen, C. and Hassan, N.U. (2021) 'Blockchain technology for energy trading and market access in renewable energy networks', in: Blockchain-Based Systems for the Modern Energy Grid. Springer, pp. 59–79.

Hicks, J. and Ison, N. (2018) 'An exploration of the boundaries of "community" in community renewable energy projects: Navigating between motivations and context', Energy Policy, 113, pp. 523–534.

IPCC (2014) Climate Change 2014: Synthesis Report. Geneva, Switzerland: Intergovernmental Panel on Climate Change.

IEA (2021) World Energy Outlook 2021. Paris: International Energy Agency.

iolite (2022) iolite webpage. (Accessed: 28 March 2024).

IRENA (2019) Innovation landscape for a renewable-powered future: Solutions to integrate variable renewables. International Renewable Energy Agency.

Jia, Y. (2018) Program Analysis Based Approaches to Ensure Security and Safety of Emerging Software Platforms. Dissertation.

Kang, J. et al. (2017) 'Enabling localized peer-to-peer electricity trading among plug-in hybrid electric vehicles using consortium blockchains', IEEE Transactions on Industrial Informatics, 13(6), pp. 3154–3164.

Khan, M.A. and Salah, K. (2018) 'IoT security: Review, blockchain solutions, and open challenges', Future Generation Computer Systems, 82, pp. 395–411.

Kiayias, A. et al. (2017) ‘Ouroboros: A provably secure proof-of-stake blockchain protocol’, in: Advances in Cryptology – CRYPTO 2017. Springer, Cham, pp. 357–388.

Kim, J. and Tong, L. (2013) ‘On topology attack of a smart grid: Undetectable attacks and countermeasures’, IEEE Journal on Selected Areas in Communications, 31(7), pp. 1294–1305.

Lamport, L., Shostak, R. and Pease, M. (1982) ‘The Byzantine Generals Problem’, ACM Transactions on Programming Languages and Systems, 4(3), pp. 382–401.

Lee, K. and Miller, A. (2018) ‘Authenticated Data Structures for Privacy-Preserving Monero Light Clients’, in: IEEE European Symposium on Security and Privacy Workshops (EuroS&PW), London, pp. 20–28.

Li, X. et al. (2017) ‘A survey on the security of blockchain systems’, Future Generation Computer Systems. doi: 10.1016/j.future.2017.08.020.

Liang, G. et al. (2018) ‘Distributed Blockchain-Based Data Protection Framework for Modern Power Systems against Cyber Attacks’, IEEE Transactions on Smart Grid, pp. 1–1. doi: 10.1109/TSG.2018.2819665.

Liang, G. et al. (2019) ‘A framework for cyber-topology attacks: Line-switching and new attack scenarios’, IEEE Transactions on Smart Grid, 10(2), pp. 1704–1712.

Liang, X. et al. (2018) ‘Towards a Reliable and Accountable Cyber Supply Chain in Energy Delivery System Using Blockchain’, in: Security and Privacy in Communication Networks, Cham: Springer, pp. 43–62.

Lisk (2022) Lisk platform webpage. (Accessed: 14 February 2024).

Liu, X., Li, Z. and Li, Z. (2017) ‘Optimal protection strategy against false data injection attacks in power systems’, IEEE Transactions on Smart Grid, 8(4), pp. 1802–1810.

London Carbon Trust (2016) An Analysis of Electricity System Flexibility for Great Britain. Carbon Trust, London.

Luke, M.N. et al. (2018) ‘Blockchain in Electricity: a Critical Review of Progress to Date’, ArXiv Preprint. arXiv:1810.09987.

Lund, H. and Mathiesen, B.V. (2009) ‘Energy system analysis of 100% renewable energy systems—The case of Denmark in years 2030 and 2050’, Energy, 34(5), pp. 524–531.

Markard, J., Raven, R. and Truffer, B. (2012) ‘Sustainability transitions: An emerging field of research and its prospects’, Research Policy, 41(6), pp. 955–967.

Mayer, R.C., Davis, J.H. and Schoorman, F.D. (1995) ‘An integrative model of organizational trust’, Academy of Management Review, 20(3), pp. 709–734.

McKinsey & Company (2022) The Net-Zero Transition: What it would cost, what it could bring.

Mengelkamp, E. et al. (2018a) ‘Designing microgrid energy markets’, Applied Energy, 210, pp. 870–880.

Mengelkamp, E. et al. (2018b) ‘A blockchain-based smart grid: Towards sustainable local energy markets’, Computer Science - Research and Development, 33(1), pp. 207–214.

Mohandes, B., Maier, M. and Kantarci, B. (2019) ‘GridSharing: Blockchain-based secure energy trading in smart grids’, IEEE Transactions on Industrial Informatics, 15(6), pp. 3330–3339.

Mylrea, M. and Gourisetti, S.N.G. (2017) ‘Blockchain for smart grid resilience: Exchanging distributed energy at speed, scale and security’, in: Resilience Week (RWS). IEEE, pp. 18–23.

Nakamoto, S. (2008) Bitcoin: A Peer-to-Peer Electronic Cash System. (Accessed: 22 March 2024).

Neblio (2016) Neblio White Paper, 26 July. (Accessed: 13 April 2024).

Octopus Energy (2021) Octopus Energy Reports. (Accessed: 04 May 2024).

Information Commissioner’s Office (ICO) (2018) Guide to the General Data Protection Regulation (GDPR).

Ofgem (2020) Flexibility Platforms in Electricity Markets.

Ofgem (2020) Facilitating Flexibility: The Role of Platforms in the Energy System.

Ofgem (2020) Data and Digitalisation.

Ofgem (2017) Future Insights Series: Flexibility.

Papavasiliou, A. and Oren, S.S. (2018) ‘Flexibility in power systems: Theory, practice, and modeling’, European Journal of Operational Research, 277(3), pp. 795–813.

European Parliament and Council of the EU (2019) Directive (EU) 2019/944 on common rules for the internal market for electricity.

UK Parliament (2008) Climate Change Act 2008.

Peck, M. (2016) ‘A blockchain currency that beats bitcoin on privacy’, IEEE Spectrum, 53(6), pp. 11–13.

Peck, M.E. (2017) ‘Blockchain world—Do you need a blockchain? This chart will tell you if the technology can solve your problem’, IEEE Spectrum, 54(10), pp. 38–60.

Peters, G.W., Panayi, E. and Chapelle, A. (2015) ‘Trends in crypto-currencies and blockchain technologies: A monetary theory and regulation perspective’, SSRN, 2646618.

Piclo (2022) Piclo Flex Annual Report 2021/22. (Accessed: 03 June 2024).

Polkadot (2017) Polkadot Lightpaper Version 1. (Accessed: 21 February 2024).

Polkadot (2024) Polkadot Network Webpage. (Accessed: 16 March 2024).

Poon, J. and Buterin, V. (2018) Plasma: Scalable Autonomous Smart Contracts - Working Draft. (Accessed: 14 April 2024).

Poon, J. and Dryja, T. (2016) The Bitcoin Lightning Network - Scalable Off-Chain Instant Payments v0.5.9.2. (Accessed: 28 January 2024).

Pope, M. and Treasure, S. (2024) Standard Technique: SD5A/6. (Accessed: 11 August 2024).

Flexible Power (2022) Flexible Power Annual Report 2021/22. (Accessed: 24 May 2024).

Public Key Cryptography (2024) Public Domain Images. (Accessed: 09 April 2024).

R3 (2024) R3 Platform Website. (Accessed: 07 March 2024).

Richardson, J. (2024) 'Peer-to-peer solar energy trading trial in Japan will use blockchain'. (Accessed: 18 May 2024).

Rilee, K. (2024) Understanding Hyperledger Sawtooth — Proof of Elapsed Time. (Accessed: 04 July 2024).

Saberi, S. et al. (2019) 'Blockchain technology and its relationships to sustainable supply chain management', International Journal of Production Research, 57(7), pp. 2117–2135.

Crown Commercial Service (2019) A Guide to Dynamic Purchasing Systems within the Public Sector. (Accessed: 23 October 2024).

Sikorski, J.J., Haughton, J. and Kraft, M. (2017) 'Blockchain technology in the chemical industry: Machine-to-machine electricity market', Applied Energy, 195, pp. 234–246.

Silvestre, M.L. et al. (2020) 'Blockchain for power systems: Current trends and future applications', Renewable and Sustainable Energy Reviews, 119, 109585.

Sousa, T. et al. (2019a) 'Peer-to-peer and community-based markets: A comprehensive review', Renewable and Sustainable Energy Reviews, 104, pp. 367–378.

Sousa, T. et al. (2019b) 'Peer-to-peer and community-based markets: A comprehensive review', Renewable and Sustainable Energy Reviews, 104, pp. 367–378.

Sovacool, B.K. and Griffiths, S. (2020) 'Culture and low-carbon energy transitions', Nature Sustainability, 3, pp. 685–693.

Swan, M. (2015) Blockchain: Blueprint for a New Economy. O'Reilly Media, Inc.

Szabo, N. (2024) Smart Contracts. (Accessed: 17 April 2024).

Szabo, N. (2024) Formalizing and Securing Relationships on Public Networks. (Accessed: 25 February 2024).

Energy Digitalisation Taskforce (2019) A Strategy for a Modern Digitalised Energy System. (Accessed: 29 March 2024).

Teotia, S. et al. (2021) 'Blockchain-based transactive energy systems: A review of recent development and future perspectives', International Journal of Energy Research, 45(1), pp. 1–21.

The Linux Foundation (2024) Hyperledger Architecture. (Accessed: 05 May 2024).

Thomas, L. et al. (2019) 'A general form of smart contract for decentralized energy systems management', Nature Energy, 4, pp. 140–149.

Energy Saving Trust (2023) Cost of Installing Renewable Technologies. (Accessed: 19 April 2024).

United Nations Framework Convention on Climate Change (UNFCCC, 2015) Paris Agreement. (Accessed: 08 October 2024).

Vélez-Ibáñez, C.G. (2010) An Impossible Living in a Transborder World: Culture, Confianza, and Economy of Mexican-Origin Populations. University of Arizona Press.

Vessenes, P. (2024) 'Ethereum Contracts Are Going to Be Candy for Hackers'. (Accessed: 20 May 2024).

Vorrath, S. (2024) 'Bitcoin-inspired peer-to-peer solar trading trial kicks off in Perth'. (Accessed: 01 June 2024).

Wang, W.Y.C., Ho, H.C.Y. and Chen, C. (2019) 'Smart contract-based settlement for energy transactions in microgrids', The Electricity Journal, 32(4), pp. 33–43.

Wang, Y., Liu, C. and Liu, J. (2019) 'An integrated energy transaction model incorporating energy currency and blockchain technology', Energies, 12(7), 1395.

Wood, D.G. (2016) Polkadot: Vision for a Heterogeneous Multi-Chain Framework, Draft 1, 21.

Wood, G. (2014) Ethereum: A Secure Decentralised Generalised Transaction Ledger. (Accessed: 06 February 2024).

Wüst, K. and Gervais, A. (2017) 'Do you need a blockchain?', in: 2017 Crypto Valley Conference on Blockchain Technology (CVCBT). IEEE, pp. 45–54.

Zhao, B. et al. (2020) 'Optimal scheduling of aggregated thermostatically controlled loads with renewable generation in the intraday electricity market', Applied Energy, 268, 114946.

Zheng, Z. et al. (2017) ‘An overview of blockchain technology: Architecture, consensus, and future trends’, in: 2017 IEEE International Congress on Big Data (BigData Congress). IEEE, pp. 557–564.

8 APPENDIX

8.1 APPENDIX A: ALGORITHM IMPLEMENTATION FOR FLEXIBILITY MARKETPLACE

8.1.1 Asset.solfz

```
//SPDX-License-Identifier: Unlicense
pragma experimental ABIEncoderV2;
pragma solidity ^0.8.4;

import "@openzeppelin/contracts/token/ERC721/ERC721.sol";
import "@openzeppelin/contracts/access/Ownable.sol";
import "@openzeppelin/contracts/access/AccessControlEnumerable.sol";
import
"@openzeppelin/contracts/token/ERC721/extensions/ERC721Enumerable.sol";
import "hardhat/console.sol";

contract Asset is ERC721, Ownable, AccessControlEnumerable,
ERC721Enumerable {

    struct AssetData {
        uint256 aec; //active export capacity
        uint256 aic; //active import capacity
        uint256 vl; // voltage level at the point of connection to the network.
        uint256 respT; // response time
        uint256 maxRuntime; //Maximum length of time that the asset can sustain
capacity.
        uint256 minRuntime; //          Minimum length of time required to dispatch the
asset.
        uint256 recT; //recovery time
        address meterAddress; //ID of the meteroit
        uint256 capacity; // response time
        uint256 expiryDate; // response time
    }
```

```
    // mapping of NFT id to its data
    mapping (uint256 => AssetData) public assetsMapping;

    // NFT counter
    uint256 nextTokenID;

    // Events
    event FlexibilityProvided(uint256 bidID, uint256 amount);
    event RequestFlexibility(uint256 bidID, uint256 ammount);
    event NoDeliveryFlexibilityBurn(uint256 bidID, uint256 ammount);

    // Able to mint new assets
    bytes32 public constant ASSETAUTH = keccak256("ASSETAUTH");

    constructor() ERC721("Asset contract", "PM 01")  AccessControl() Ownable() {

                _setupRole(DEFAULT_ADMIN_ROLE, msg.sender);  // set up mega-
admin account
                grantRole(ASSETAUTH, msg.sender);                    // spawner has
privillages to add more adders
        }

    // // need to timeLock the asset if won bid
    // function deliverFlex(uint256 _bidID, uint256 _amount) external {
    //     emit FlexibilityProvided(_bidID, _amount);
    // }

    // function noDeliverFlexBurn(uint256 _bidID, uint256 _amount) external {
    //     emit NoDeliveryFlexibilityBurn(_bidID, _amount);
    // }

    function addAsset(
        AssetData memory _assetData,
        address _assetOwner
```

```
    ) external onlyRole(ASSETAUTH) returns (uint256) {

        // Mint asset NFT
        _safeMint(_assetOwner, nextTokenID);

        // Set asset data in mapping
        assetsMapping[nextTokenID] = _assetData;

        // Increment NFT ID counter
        nextTokenID++;

        return nextTokenID-1;
    }


    function updateExpiryDate(uint256 _assetID, uint256 _newExpiry ) external
onlyRole(ASSETAUTH) {

        // TODO: require timecheck to be in the future
        assetsMapping[_assetID].expiryDate = _newExpiry;
    }


    //  function getMilestone(uint n) public constant returns (uint, uint ) {
    //     return (milestones[n].time, milestones[n].price);
    // }

    function transferFrom(
        address from,
        address to,
        uint256 tokenId
    ) public override onlyRole(ASSETAUTH){
        super.transferFrom(from, to, tokenId);
    }
```

```
    function safeTransferFrom(
        address from,
        address to,
        uint256 tokenId,
        bytes memory _data
    ) public override onlyRole(ASSETAUTH){
        super.safeTransferFrom(from, to, tokenId, _data);
    }



    function _beforeTokenTransfer(
                address from,
                address to,
                uint256 tokenId
        ) internal override(ERC721, ERC721Enumerable) {
                super._beforeTokenTransfer(from, to, tokenId);
        }


    function supportsInterface(bytes4 interfaceId)
                public
                view
                override(ERC721, AccessControlEnumerable, ERC721Enumerable)
                returns (bool)
        {
                return super.supportsInterface(interfaceId);
        }


}
```

8.1.2 Confidential Asset.sol

```
//SPDX-License-Identifier: Unlicense
```

```
pragma solidity ^0.8.4;

import "@openzeppelin/contracts/token/ERC721/ERC721.sol";

import "hardhat/console.sol";

// ERC721 representation of a physical facility capable
// of delivering energy on demand to the grid.
// All parameters besides export and import capacity are
// only known to to the DSO submitting contract
contract ConfidentialAsset is ERC721 {

    uint256 aec; //active export capacity
    uint256 aic; //active import capacity

    event FlexibilityProvided(uint256 bidID, uint256 amount);
    event RequestFlexibility(uint256 bidID, uint256 ammount);
    event NoDeliveryFlexibilityBurn(uint256 bidID, uint256 ammount);

    constructor(
        uint256 _aec, //active export capacity
        uint256 _aic //active import capacity
    ) ERC721("Foo", "foo")  {
        aec = _aec; //active export capacity
        aic = _aic; //active import capacity
    }

    function requestFlex(uint256 _bidId, uint256 _ammount)  external
        returns (string memory)
    {
        emit RequestFlexibility(_bidId, _ammount);
        return "Deliver BITCH";
    }
```

```
    // need to timeLock the asset if won bid
    function deliverFlex(uint256 _bidID, uint256 _amount) external {
        // flexCoin.deliverFlexCoin(_amount);
        emit FlexibilityProvided(_bidID, _amount);
    }

    function noDeliverFlexBurn(uint256 _bidID, uint256 _amount) external {
        // flexCoin.deliverFlexCoin(_amount);
        emit NoDeliveryFlexibilityBurn(_bidID, _amount);
    }
}
```

8.1.3 Escrow.sol

```
// //SPDX-License-Identifier: Unlicense
// pragma solidity ^0.8.4;

// import "./Asset.sol";
// import "./IERC20.sol";
// import "@openzeppelin/contracts/security/ReentrancyGuard.sol";



// contract Escrow is ReentrancyGuard {

//     // Callers are instances of FlexContract
//     event Deposited(address indexed payee, uint256 amountStablecoin);
//     event Withdrawn(address indexed payee, uint256 amountStablecoin);

//     address addressStablecoin;
//     IERC20 stablecoin;
//     uint256 stabeCoinDecimals;

//     // Deposit instance is made up of
//     // [Total funds, Deposit held in this contrac, Deposit plugged into Defi]
//     // mapping(address => uint256) private _deposits;
```

```
//     // Big boys TODO:
//     constructor(address _addressStablecoin){
//         addressStablecoin = addressStablecoin;
//         stablecoin = IERC20(addressStablecoin);     // Set stable coin to use for
deposits, withdraws and yield
//     }

//     function depositsOf(address payee) public view returns (uint256) {
//         return _deposits[payee];
//     }


//     // Called from flexibility contract
//     function depositDeFi() external nonReentrant{

//         // Take this and plug into aave


//     }

//     // Get all the profit and withdraw
//     function withdrawProfit(address payee) external nonReentrant {

//     }

//     function deposit(address payee, uint256 depositAmount) external nonReentrant
{

//         // Check approval mathces amunt dolls
//         uint256 spendAllowance = stablecoin.allowance(payee, address(this));

//         // Can deposit
//         require(depositAmount <= spendAllowance);
```

```
//         // Deposit stablecoin
//         _deposits[payee] += depositAmount;

//         // Transfer to here
//         stablecoin.transfer(address(this), depositAmount);
//         emit Deposited(payee, depositAmount);
//     }

//     function withdraw(address withdrawer, uint256 withdrawAmount) external
nonReentrant {

//         // Get balance from mapping
//         uint withdrawerBalance = _deposits[withdrawer];

//         // Withdraws only what it owns
//         require(withdrawerBalance >= withdrawAmount);

//         // Update new amount
//         _deposits[withdrawer] = withdrawerBalance - withdrawAmount;

//         // Withdraw stablecoin
//         stablecoin.transfer(withdrawer, withdrawAmount);
//         emit Withdrawn(withdrawer, withdrawAmount);
//     }
// }
```

8.1.4 FlexContract.sol

```
//SPDX-License-Identifier: Unlicense
pragma solidity ^0.8.4;

import "./Asset.sol";
import "./IERC20.sol";
import "./IAavePool.sol";
import "@openzeppelin/contracts/security/ReentrancyGuard.sol";
```

```
import "./IERC20.sol";
import "./IAavePool.sol";

contract FlexContract is ReentrancyGuard {
    address public admin; // end and keep the profit
    address addressStablecoin;
    address stablecoinAddress;
    address LPAddress;
    address aavePoolAddress;

    IERC20 stablecoin; // Stale coing for De-Fi
    IERC20 aToken;
    IAavePool pool;
    uint256 erc20_decimal;
    bool public noMoreWindows; // Signifies end of window setting
    bool public isTypeActive; //T-Active F-Reactive

    enum FlexServiceType {
        SECURE,
        DYNAMIC,
        RESTORE,
        SUSTAIN
    }

    enum ServiceDeliverd {
        STANDBY,
        DELIVERY,
        FAILURE
    }

    struct ServiceWindow {
        uint256 swStart; // Start time of the day for the flex required. If the window
length is all day,
```

```
        uint256 swEnd; // End time of the day for the flex required. If the window is all
day,
        // uint256[] serviceDays; // What days of the week the flexibility is required for.
        uint256 capReq; // How many MW or MVAr are required in that particular
window. Unit is defined by competition Power Type.
        // uint256 minAggAssSize; // The minimum total aggregate capacity of a group
of assets that can qualify for bidding (in MW or MVar)
        uint256 minRunTime; // Minimum time in minutes required of the asset to
provide flexibility. Leave blank to require continuous/constant operation.
        uint256 reqRespTime; // Time within which an asset must respond to a
utilisation request.
        uint256 dispatchDuration; //The estimated duration of each dispatch event.
        uint256 dispatchEst; // The estimated number of dispatch events expected
during the service period of this service window.
        // uint256 AvFinalPrice; //Final amount paid to Flex Providers to be available
        // uint256[2] utilityTimeStamps;
        uint256 supplyStart;
        uint256 supplyEnd;
        uint256 UtFinalPrice; //Final amount paid to Flex Providers when flexibility is
utilised
        uint256 ServFinalFee; //Final amount paid to Flex Providers for capacity
        ServiceDeliverd jobDone;
    }

    struct Bid {
        address bidOwner; // owner of the asset
        address meterAddress; // meter at the asset
        uint256 capacity;
        uint256 maxRuntime;
        uint256 avPrice;
        uint256 utPrice;
        uint256 servicePrice;
        uint256 serviceWindowId;
    }
```

```
    uint256 public counterSW; // How many windows within this contract
    uint256 public counterBids; // How many bids submitted

    mapping(uint256 => ServiceWindow) public idServiceWindows;
    mapping(uint256 => Bid) public bidIds;
    mapping(uint256 => uint256[]) public bidsForWindow; // [0] => [1,2]
    mapping(uint256 => uint256) public bidsWonWindow; // [0] => [2]

    // Events
    event ServiceWindowCreated(uint256 serviceWindowId); // Emmit for adding a
new window
    event BiddingOn(); // Emmit for bidding is open
    event Deposit(address payer, uint256 amount); // Emmit for bidding is open
    event Withdrew(address payer, uint256 amount); // Emmit for bidding is open
    event WindowStart(address meterID); // Emmit when window starts
    event WindowWinner(address meterID); // Emmit when window starts
    event WindowEnd(address meterID); // Emmit when window starts

    struct ContractCriteria {
        uint256 minConVoltage;
        uint256 maxConVoltage;
        FlexServiceType productType;
        uint256 cmzOpen;
        uint256 cmzClose;
        bool isFixedPrice;
        uint256 minBudget;
        uint256 maxBugdet;
        uint256 avGuidePrice;
        uint256 utGuidePrice;
        uint256 serviceFee;
    }

    ContractCriteria public contractCriteria;
```

```
    constructor(
        address _stablecoin,
        address _LPAddress,
        address _aavePoolAddress,
        ContractCriteria memory _contractCriteria
    )
    {
        stablecoin = IERC20(_stablecoin); // Set stable coin to use for deposits and
payments
        erc20_decimal = stablecoin.decimals(); // Set decimal point of the stablecoin
        aToken = IERC20(LPAddress);
        pool = IAavePool(_aavePoolAddress);
        console.log("Address of the stablecoin %s", _stablecoin);
        admin = msg.sender; // Authority
        contractCriteria = _contractCriteria;
        counterSW++; // Cheap increment
    }

    modifier onlyAdmin() {
        require(msg.sender == admin, "Caller is not an admin");
        _;
    }

    // DNO adds serice window
    function addServiceWindow(ServiceWindow memory _data) external onlyAdmin {
        // Check allowance by the DNO
        uint256 allowanceWindowDNO = stablecoin.allowance(
            msg.sender,
            address(this)
        );

        // Allowance larger than utGuidePrice
        require(_data.UtFinalPrice <= allowanceWindowDNO);
```

```
    // Put window requirements into the mapping based on counter/index
    idServiceWindows[counterSW] = _data;

    // Transfer deposit from flexibility provider to AAVE
    stablecoin.transfer(msg.sender, allowanceWindowDNO);

    emit ServiceWindowCreated(counterSW);
    counterSW++;
}

// DNO signals to smart meter that a window is starting
function reportStartWindow(uint256 windowID) external onlyAdmin {
    // Retrieve address of the asset that is to start
    address meterAddress = bidIds[bidsWonWindow[windowID]].meterAddress;

    // Get block time for start of the window
    idServiceWindows[windowID].supplyStart = block.timestamp;

    // Emit event stating the address of the pi
    emit WindowStart(meterAddress); // Emmit when window starts
}

// DNO signals that a window is ending
function reportEndWindow(uint256 windowID) external onlyAdmin {
    // Retrieve address of the asset that is to start
    // bidIds[bidsWonWindow[windowID]].meterAddress
    address meterAddress = bidIds[bidsWonWindow[windowID]].meterAddress;

    // Get block time for end of the window
    idServiceWindows[windowID].supplyEnd = block.timestamp;

    // Emit event stating the address of the pi
    emit WindowEnd(meterAddress); // Emmit when window starts
```

```
}

// Pi reports job done
function reportServiceWindow(uint256 windowID, ServiceDeliverd sd)
    external
{
    // Ensure the correct meter only can notify of job performence
    require(
        msg.sender == bidIds[bidsWonWindow[windowID]].meterAddress,
        "Caller is not smart meter"
    );

    // Set the job rating
    idServiceWindows[windowID].jobDone = sd;

    // Calculate the fee based on uptime

    // event WindowEnd(address meterID); // Emmit when window starts
}

// Refund deposits

function endWindows() external onlyAdmin {
    noMoreWindows = true;
    emit BiddingOn();
}

// Submission by an ERC721 asset
function submitBid(
    address _factoryNFT,
    uint256 _assetID,
    Bid memory _bid
) external {
    // Require windows to be finalised
```

```
        require(noMoreWindows == true, "Windows not finalised");

        // Rerieve window requirments
        ServiceWindow memory serviceWindow =
idServiceWindows[_bid.serviceWindowId];

        // Get instance of the NFT
        Asset assetFactory = Asset(address(_factoryNFT));

        // Authority check
        require(assetFactory.ownerOf(_assetID) == msg.sender);

        // Valid timewindow check
        require(contractCriteria.cmzOpen > block.timestamp, "Window expired");
//check if the service window is opened TODO: windowing

        // Get capacity from the bidder
        (, , , , , , , uint256 capacity, uint256 expiryDate) = assetFactory
            .assetsMapping(_assetID);

        // Capacity check
        require(
            capacity >= serviceWindow.capReq,
            "Bid's capacity is too small"
        );

        // Certificate validity check
        require(expiryDate >= block.timestamp, "DPS not registed or expired");

        // Deposit amount based on 10% of the one provided by the DNO for this
window
        uint256 windowDeposit = serviceWindow.UtFinalPrice / 10;

        // Allowance to transfer to flexcontract larger than utGuidePrice
```

```
    require(windowDeposit <= stablecoin.allowance(msg.sender, address(this)));

    // Transfer deposit from flexibility provider to flexcontract
    stablecoin.transferFrom(msg.sender, address(this), windowDeposit);

    // Give allowance from flexcontract to aave pool
    stablecoin.approve(aavePoolAddress, windowDeposit);

    // Deposit deposit to aave pool
    pool.deposit(
        stablecoinAddress,
        windowDeposit,
        msg.sender,
        0
    );

    // Submit bid to the total bid mapping
    bidIds[counterBids] = _bid;

    // Submit bid to the given window mapping mapping
    bidsForWindow[_bid.serviceWindowId].push(counterBids);
    counterBids++;

    // Event for bid submission
    emit Deposit(msg.sender, windowDeposit);
}

function payout(
    uint256 _bidID,
    address _factoryNFT,
    uint256 _assetID
) external {

    // Payout sum
```

```
uint256 payoutSum = 0;

// Get instance of the NFT
Asset assetFactory = Asset(address(_factoryNFT));

// Authority check
require(assetFactory.ownerOf(_assetID) == msg.sender);

// Get the window this asset bid for
uint256 windowId = bidIds[_bidID].serviceWindowId;

// Check if asset won the window
if (bidsWonWindow[windowId] == _bidID){

    // If they didn't deliver return nothing
    if (idServiceWindows[windowId].jobDone == ServiceDeliverd.FAILURE){
        return;
    }

    // Add deposit
    payoutSum += idServiceWindows[windowId].UtFinalPrice / 10;

    if (idServiceWindows[windowId].jobDone == ServiceDeliverd.DELIVERY){

        // Add fee
        payoutSum += idServiceWindows[windowId].UtFinalPrice;
    }
}

// Approve bid placer
aToken.approve(assetFactory.ownerOf(_assetID), payoutSum);
pool.withdraw(
    stablecoinAddress,
    payoutSum,
```

```
        assetFactory.ownerOf(_assetID)
    );

    emit Withdrew(msg.sender, payoutSum);
}

function updateWindowWinner(uint256 _serviceWindowID, uint256 _winnerBidID)
    external
    onlyAdmin
{
    // Require windows to be finalised
    require(noMoreWindows == true, "Windows not finalised");

    // Require at least one bid
    require(counterBids > 0, "No bids submitted");

    bidsWonWindow[_serviceWindowID] = _winnerBidID;
    // tranfer tokens for SW to...
    // uint256 index;
    // assembly {
    //     index := mload(add(_winnerBidID, 0))
    // }
    // for (uint256 ii; ii < index; ii++) {
    //     Bid memory winnerBid = bidIds[_winnerBidID[ii]];
    // }

    emit WindowWinner(bidIds[bidsWonWindow[_winnerBidID]].meterAddress);
}

function withdrawProfit(uint256 _amount, uint256 _recipient)  external  {

}

function changeAdmin(address _newAdmin) external onlyAdmin {
```

```
        admin = _newAdmin;
    }
}
```

8.2 APPENDIX B: ALGORITHM IMPLEMENTATION FOR ZKP FOR FLEXIBILITY SETTLEMENT

8.2.1 Asset.ts

```
/
- Asset details:
        Fields:
            - Owner's public key
            - Link to physical attributes (location, power, certification)
        Functions:
            - updateOwner():
            - updateLinkSpecs():

 */
import {
  Field,
  SmartContract,
  state,
  State,
  method,
  PublicKey,
  UInt64,
  Provable,
} from 'snarkyjs';
import { Flex } from './Flex';

export class Asset extends SmartContract {
  @state(PublicKey) owner = State<PublicKey>();
  @state(UInt64) powerCapacity = State<UInt64>();
  @state(PublicKey) approvedSigner = State<PublicKey>();
```

```
// @state(Field) hashOfDetails = State<Field>();

init() {
  super.init();
  this.owner.set(this.sender);
  this.account.zkappUri.set(
    'www.dnoSecureWebsite.com/asset/' + this.address.toBase58()
  );
}

// Would pass struct of details but save only hash
@method setup(
  newOwner: PublicKey,
  powerCapacity: UInt64,
  approvedSigner: PublicKey
) {
  const owner = this.owner.get();
  this.owner.assertEquals(owner);

  owner.assertEquals(this.sender);
  this.owner.set(newOwner);
  this.powerCapacity.set(powerCapacity);
  this.approvedSigner.set(approvedSigner);
}

//only owner
@method bidOnContract(flexContractAddress: PublicKey, pricePerKw: UInt64) {
  const owner = this.owner.get();
  this.owner.assertEquals(owner);
  owner.assertEquals(this.sender);

  const flexContract = new Flex(flexContractAddress);
  flexContract.bid(this.address, pricePerKw);
}
```

```
}
```

8.2.2 Flex.ts

```
/
 - Flexibility contract (escrows money): On chain reference for settlement
        Fields:
                - dateTime    !
                - power                ! kW
  ---------------------------
                - dispatch time
                - flexibility service
                - prices
                - DNO Key
                - Flexibility contract agreed flag
                - Delivered flag
 */

import {
  Field,
  SmartContract,
  state,
  State,
  method,
  PublicKey,
  UInt64,
  Reducer,
  Struct,
  Provable,
  Bool,
  Signature,
} from 'snarkyjs';
import { Factory } from './Factory';
import { Asset } from './Asset';
```

```
import { READINGS, MINUTE, THIRTY_MINUTES, TWENTY_SECONDS } from
'./utils';

export class Delivery extends Struct({
  signature: Signature,
  kwh: Provable.Array(Field, READINGS),
  timestamp: Provable.Array(Field, READINGS),
}) {}
// class BidAction extends Struct({ address: PublicKey, pricePerKw: UInt64 }) {}
export class Flex extends SmartContract {
  // @state(PublicKey) factoryContract = State<PublicKey>();
  //MIn capacity ?
  @state(UInt64) startTime = State<UInt64>();
  @state(UInt64) endTime = State<UInt64>();
  @state(UInt64) powerLeftToDeliver = State<UInt64>();
  @state(PublicKey) winner = State<PublicKey>();
  @state(PublicKey) admin = State<PublicKey>();
  @state(Field) submissionCounter = State<Field>();

  // helper field to store the point in the action history that our on-chain state is at
  // @state(Field) actionState = State<Field>();
  // reducer = Reducer({ actionType: BidAction });

  events = {
    bidMade: Struct({ asset: PublicKey, pricePerKw: UInt64 }),
    winningAsset: PublicKey,
  };

  init() {
    super.init();

    this.account.zkappUri.set(
      'www.dnoSecureWebsite.com/contract/' + this.address.toBase58()
    );
```

```
  this.winner.set(this.address);
  //test above
  // this.actionState.set(Reducer.initialActionState);
}

@method setup(
  factoryContract: PublicKey, //TODO needed ?
  startTime: UInt64,
  endTime: UInt64,
  contractedWatts: UInt64,
  admin: PublicKey
) {
  this.admin.set(admin);
  // this.factoryContract.set(factoryContract);
  this.startTime.set(startTime);
  this.endTime.set(endTime);
  this.powerLeftToDeliver.set(contractedWatts);
}

@method bid(assetAddress: PublicKey, pricePerKw: UInt64) {
  const winner = this.winner.get();
  this.winner.assertEquals(winner);
  //make sure winner is not set
  winner.assertEquals(this.address);
  //make sure it's called by asset contract
  const asset = new Asset(assetAddress);
  // asset.owner.assertEquals(this.sender);
  let assetCapacity = asset.powerCapacity.get();
  //  // verify min requiremnts !
  // assetCapacity.assertGreaterThanOrEqual;

  // reducer
  // this.reducer.dispatch({ address: this.sender, pricePerKw });
  this.emitEvent('bidMade', { asset: assetAddress, pricePerKw });
```

```
}
@method setWinner(winner: PublicKey) {
  //only admin
  const admin = this.admin.get();
  this.admin.assertEquals(admin);
  admin.assertEquals(this.sender);

  this.winner.set(winner);
  this.emitEvent('winningAsset', winner);
}

@method deliveryUpdate(delivery: Delivery) {
  const winningAsset = this.winner.get();
  this.winner.assertEquals(winningAsset);
  // Get permitted device
  const asset = new Asset(winningAsset);
  const approvedSigner = asset.approvedSigner.get();
  asset.approvedSigner.assertEquals(approvedSigner);

  // Get start, end times and index
  let startTime = this.startTime.get();
  this.startTime.assertEquals(startTime);
  let endTime = this.endTime.get();
  this.endTime.assertEquals(endTime);
  let submissionCounter = this.submissionCounter.get();
  this.submissionCounter.assertEquals(submissionCounter);

  // Get power left to deliver
  let powerLeftToDeliver = this.powerLeftToDeliver.get();
  this.powerLeftToDeliver.assertEquals(powerLeftToDeliver);

  // Block offset
  let blockOffset = UInt64.from(
    UInt64.from(startTime).add(
```

```
    UInt64.from(THIRTY_MINUTES).mul(UInt64.from(submissionCounter))
  )
);

let minuteOffset = blockOffset;

// Start limits
let limitHigh = UInt64.from(minuteOffset).add(UInt64.from(TWENTY_SECONDS));
let limitLow = UInt64.from(minuteOffset).sub(UInt64.from(TWENTY_SECONDS));

let remainder: UInt64;
//30 min in a submission window
for (let i = 0; i < 30; i++) {
  // Assert within the permitted range
  UInt64.from(delivery.timestamp[i]).assertLessThanOrEqual(limitHigh);
  UInt64.from(delivery.timestamp[i]).assertGreaterThanOrEqual(limitLow);

  // Add minute since last minute offset
  minuteOffset = minuteOffset.add(UInt64.from(MINUTE));

  // Increment offset and limits
  limitHigh = minuteOffset.add(UInt64.from(TWENTY_SECONDS));
  limitLow = minuteOffset.sub(UInt64.from(TWENTY_SECONDS));

  // Delivered amount exceeds minimal requirements
  UInt64.from(delivery.kwh[i]).assertGreaterThanOrEqual(UInt64.from(2));

  // Ensure it doesn't go below zero
  remainder = Provable.if(
    powerLeftToDeliver.lessThanOrEqual(UInt64.from(delivery.kwh[i])),
    powerLeftToDeliver,
    UInt64.from(delivery.kwh[i])
  );
```

```
  // Subtract
  powerLeftToDeliver = powerLeftToDeliver.sub(remainder);
}

// Check signature over the whole thing
delivery.signature
  .verify(approvedSigner, [...delivery.kwh, ...delivery.timestamp])
  .assertTrue();

// this.submissionCounter.assertEquals(submissionCounter);
this.submissionCounter.set(submissionCounter.add(Field(1)));

// Set new power level
this.powerLeftToDeliver.set(powerLeftToDeliver);

// 'choose cheapest bid'
// @method closeContract() {
//   //rollup reducer
//   //assert winner dominance
//   let actionState = this.actionState.get();
//   this.actionState.assertEquals(actionState);
//   let winner = this.winner.get();
//   this.winner.assertEquals(winner);
//   let winningAddr: PublicKey;
//   // compute the new counter and hash from pending actions
//   let pendingActions = this.reducer.getActions({
//     fromActionState: actionState,
//   });

//   let { state: newWinner, actionState: newActionState } = this.reducer.reduce(
//     pendingActions,
//     // state type
//     BidAction,
//     // function that says how to apply an action
```

```
    //     (state: BidAction, action: BidAction) => {
    //       return Provable.if(
    //         action.pricePerKw.greaterThan(state.pricePerKw),
    //         BidAction,
    //         action,
    //         state
    //       );
    //     },
    //     { state: winningAddr, actionState }
    //   );

    //   // update on-chain state
    //   this.winner.set(newWinner.address);
    //   this.actionState.set(newActionState);
    // }
  }
}
```

8.3 APPENDIX C: ALGORITHM IMPLEMENTATION FOR ROSCA-BASED FINANCING TOOL FOR FLEXIBILITY

8.3.1 GroupOwner.sol

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;

contract GroupOwnerContract {
    address public groupOwner;
    uint256 public contributionAmount;
    uint256 public totalCycles;
    uint256 public currentCycle;
    uint256 public groupSize;
    address[] public participants;
    bool public registrationOpen;

    mapping(address => bool) public isRegistered;

    constructor(uint256 _contributionAmount, uint256 _totalCycles, uint256
_groupSize) {
        groupOwner = msg.sender;
        contributionAmount = _contributionAmount;
        totalCycles = _totalCycles;
        groupSize = _groupSize;
        registrationOpen = true;
    }

    function register() external {
        require(registrationOpen, "Registration is closed");
        require(!isRegistered[msg.sender], "Already registered");
        require(participants.length < groupSize, "Group is full");

        isRegistered[msg.sender] = true;
        participants.push(msg.sender);
```

```
        if (participants.length == groupSize) {
            registrationOpen = false;
        }
    }

    function setAsset(uint256 assetValue) external {
        require(msg.sender == groupOwner, "Only group owner can set asset");
        // Logic to set the asset for the group
    }

    function startCycle() external {
        require(msg.sender == groupOwner, "Only group owner can start cycle");
        currentCycle++;
    }

    // Further functions for managing raffles, contributions, and settlements
}
```

8.3.2 ContributionManagement.sol

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;

contract ContributionManagementContract {
    address public groupOwner;
    uint256 public contributionAmount;
    mapping(address => uint256) public contributions;
    mapping(address => uint256) public lastContributionCycle;

    modifier onlyOwner() {
        require(msg.sender == groupOwner, "Not the group owner");
        _;
    }
```

```
    constructor(uint256 _contributionAmount) {
        groupOwner = msg.sender;
        contributionAmount = _contributionAmount;
    }

    function contribute() external payable {
        require(msg.value == contributionAmount, "Invalid contribution amount");
        contributions[msg.sender] += msg.value;
        lastContributionCycle[msg.sender]++;
    }

    function checkContribution(address participant) external view returns (uint256) {
        return contributions[participant];
    }

    // Further functions for managing penalties, missed contributions, and fund locking
}
```

8.3.3 Auction.sol

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;

contract AuctionContract {
    address public highestBidder;
    uint256 public highestBid;
    address[] public participants;
    bool public auctionOpen;

    constructor() {
        auctionOpen = true;
    }

    function placeBid() external payable {
        require(auctionOpen, "Auction closed");
```

```
        require(msg.value > highestBid, "Bid is too low");

        if (highestBidder != address(0)) {
            payable(highestBidder).transfer(highestBid);  // Refund previous highest bid
        }

        highestBidder = msg.sender;
        highestBid = msg.value;
    }

    function closeAuction() external {
        require(auctionOpen, "Auction already closed");
        auctionOpen = false;
        // Transfer the asset to the highestBidder
    }
}
```

8.3.4 Raffle.sol

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;

contract RaffleContract {
    address[] public participants;
    bool public raffleOpen;

    constructor() {
        raffleOpen = true;
    }

    function enterRaffle() external {
        require(raffleOpen, "Raffle is closed");
        participants.push(msg.sender);
    }
```

```
    function drawWinner() external view returns (address) {
        require(raffleOpen, "Raffle closed");
        require(participants.length > 0, "No participants");

        // Simple random selection using block information
        uint256 randomIndex = uint256(keccak256(abi.encodePacked(block.timestamp,
block.difficulty))) % participants.length;
        return participants[randomIndex];
    }

    function closeRaffle() external {
        raffleOpen = false;
        // Asset transfer logic to the winner
    }
}
```

8.3.5 Settlement.sol

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;

contract SettlementContract {
    address public groupOwner;
    mapping(address => bool) public assetsTransferred;

    modifier onlyOwner() {
        require(msg.sender == groupOwner, "Only group owner can execute");
        _;
    }

    constructor() {
        groupOwner = msg.sender;
    }

    function settleAsset(address participant) external onlyOwner {
```

```
        require(!assetsTransferred[participant], "Asset already transferred");

        // Logic for transferring digital assets directly or working with oracles
        assetsTransferred[participant] = true;
    }

    function settleFunds(address participant, uint256 amount) external onlyOwner {
        payable(participant).transfer(amount);  // Transfer funds back to the participant
    }
}
```