Analysis and Design of Scalable Blockchain-based Smart Contract System for Smart Grid Monitoring and Control

by

Kimia Honari

A thesis submitted in partial fulfillment of the requirements for the degree of

Master of Science

in

Software Engineering and Intelligent Systems

Department of Electrical and Computer Engineering
University of Alberta

Abstract

Energy systems are undergoing rapid changes to adapt to both increasing demand and growing penetration of embedded renewable generation. The Smart Grid, which integrates information and communication technologies into the grid infrastructure, is one possible solution to both needs. However, if the number of residential prosumers grows to a significant fraction of the total customer base, the current, centralized systems for managing energy markets and operations will be insufficient. New, distributed systems will be needed instead. Blockchains, which are a peer-to-peer (P2P) decentralized ledger technology, and smart contracts built upon them, appear to be a promising approach to designing these systems. This is a fairly new area of investigation, and the literature to date is largely fragmented. Hence, in the first part of this thesis, we conduct a systematic review of distributed energy management through blockchain technology, focusing on smart contract design and development. We categorize the application domains into four main fields, including market operations, ancillary services, auditing and monitoring, and cybersecurity. We determined that data storage and blockchain interoperability are cross-cutting concerns in all of these areas, and we examined solutions for them.

Renewable energy sources in the energy system produce power intermittently, depending on weather conditions. This raises new challenges in the management and operation of electricity system, as intermittency negatively impacts the existing control measures used to ensure safe operation and stability. Furthermore, the future energy market may well include tens or hundreds of thousands of individual actors in the markets that will sell power from renewable and micro-generation. All

these complexities and challenges in the grid are putting increasing pressure on the monitoring and control systems, in particular voltage stability control. Hence, decentralized voltage stability algorithms are receiving considerable attention. This class of algorithm principally operates on localized voltage measurements but still ensures system-level stability. Several studies have used blockchains to provide ancillary services by tracking and managing energy distribution or organizing some distributed energy resources. However, the performance of blockchain-based systems in real-time grid monitoring and control has never been empirically tested. In the second part of this thesis, we propose implementing a decentralized voltage stability algorithm, using blockchain-based smart contracts, as a testbed for evaluating the performance of blockchains in real-time control. We furthermore investigate sharding mechanisms as a means of improving the system's scalability with fixed computing resources. We implement our models as a proof-of-concept prototype system using Hyperledger Fabric as our blockchain platform, the Matpower library in MATLAB as our power system simulator, and Hyperledger Caliper as our performance evaluation tool. We found that sharding does indeed lead to a substantial improvement in system scalability for this domain, measured by both transaction success rates and transaction latency.

Preface

The research of this thesis has been conducted under the supervision of Dr. Scott Dick and Dr. James Miller.

Chapter 3 is a collaboration with Sara Rouhani, assistant professor at the University of Manitoba, and Nida E. Falak, former MEng student at the University of Alberta. This work was conducted under the supervision of Dr. Scott Dick and Dr. Hao Liang. Parts of this work have been submitted as "Smart contract design in distributed energy systems: a systematic review" in IEEE Transactions on Smart Grid. I was responsible for summarizing almost half of the gathered papers, classifying and organizing all papers, developing half of the manuscript, and creating and organizing all tables and semantic network diagrams.

Parts of Chapter 4 has been accepted as a regular paper in IEEE Blockchain Conference 2022 [1][1]: K. Honari, X. Zhou, S. Rouhani, S. Dick, H. Liang, YW. Li, J. Miller, "A Scalable Blockchain-based Smart Contract Model for Decentralized Voltage Stability Using Sharding Technique." I was responsible for implementing algorithms, preparing the blockchain and sharding testbed, conducting the experiments, and developing the manuscript.

[1] http://www.blockchain-ieee.org/

Acknowledgements

First and foremost, I would like to express my sincere gratitude to my supervisor, Dr. Scott Dick, for his support and guidance throughout my graduate studies. It has been a privilege for me to work under the supervision of such a knowledgeable, kind, and energetic professor.

I would like to thank Dr. Hao Liang and Xiaotian Zhou for their guidance and assistance in understanding and implementing energy grid systems and market operations. I also would like to thank Sara Rouhani, assistant professor at the University of Manitoba, who kindly helped and guided me through these projects.

Last but by no means least, I would like to thank my parents for their unconditional love, support, and encouragement.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

Electricity demand has increased constantly for decades. Furthermore, the global response to climate change hinges on shifting energy consumption largely to cleanly-produced electricity rather than burning fossil fuels to produce heat or mechanical force [2]. The rising adoption of electric vehicles is only one facet of the energy transition (HVAC electrification is likely to be even more demanding), and so the demand for reliable electric power continues to rise. The global Smart Grid technology market is estimated to be USD 53.14 billion in 2022 and is projected to reach USD 117.21 billion by 2027 [3]. It's estimated that North American utilities will need to construct between $1.5 to $2 trillion in new generation capacity and Smart Grids by 2030 [4] to meet the twin demands of surging demand and climate change. The Smart Grid, which leverages Information Technology (IT) to improve energy system efficiencies, is a key response to these needs. The Smart Grid is also essential for increasing the penetration of renewable generation and enabling the effective management and distribution of renewable energy sources such as solar, wind, and hydrogen [5]. According to the International Energy Agency (IEA), Canada has put the country on a path toward transforming its energy system by cutting greenhouse gas emissions by about 40-45% by 2030 and reaching net-zero emissions by 2050 [6][2]. The first steps in deploying the Smart Grid have now been taken. Advanced Metering Infrastructure

(AMI) is being widely deployed, and various jurisdictions are exploring time-of-use pricing, large-scale renewables integration, etc. In the United States of America, investments in smart grid technology in 2018 amounted to $6.4 billion and are forecast to grow to $16.4 billion annually by 2026 [5].

However, climate change affects both wind and solar power generation as well as electricity demand [7] [8]. On the demand side, the balance between heating and cooling demand patterns is changing due to rising temperatures. On the supply side, changing the distribution and variability of wind, solar and hydropower resources affects the physical plants and market models that extract and deliver that energy. Other impacts include reduced water flows, and thus cooling capacity, at some power plants, forcing a reduction in output. Furthermore, the increasing popularity of electric vehicles (EVs) may necessitate charging millions of EVs in a single night; an immense new load upon generation, transmission and distribution systems [9]. In addition, the future energy market will include tens or hundreds of thousands of individual actors in the markets [5]. A multitude of entities will be able to develop renewable and micro-generation (RDM generation); from solar roofs and windmills installed by residential customers to community-scale solar/wind power co-ops, large-scale distributed generation (e.g. diesel or LNG generators in the far North), right through to the largest solar farms. The intermittency of renewables, and the unpredictable and unscheduled power flows from massive numbers of distributed or micro-generation sites, will all be interacting in real-time. It seems clear that the current information systems in power markets will simply not scale up to the Smart Grid's future needs, which may significantly compromise voltage stability.

The existing energy supply system [5]- such as how a transaction between generators and consumers is conducted, verified and recorded - is already hugely complex. Meanwhile, the deployment of Intelligent Electronic Devices (lEDs) and Phasor Measurement Units (PMUs) - meant to enable fine-grained monitoring and control of the grid - has in turn created a tsunami of sensor data that threatens to overwhelm

centralized control systems [5]. Considering all of these factors, a new, highly scalable and decentralized IT platform will be needed in order to coordinate all of these entities and data transactions. Otherwise, it is likely to become increasingly difficult to maintain equilibrium in energy markets and voltage stability and power quality on the grid.

1.2 Research Objectives

Blockchain is a distributed shared ledger replicated across all nodes communicating through a peer-to-peer network. In the blockchain, data is added to the ledger as a group of transactions called blocks. Every block is linked to the previous block by including the previous block's hash value in the header of the current block. Because of their unique data structure, blockchains offer desirable features including immutability, tamper resistance and data provenance. Blockchains also employ consensus mechanisms to reach an agreement on the new state of the blockchain, validate transactions, and ensure randomness in selecting block validators. Within a blockchain, smart contracts are executable code that implements the logic behind each transaction and can be employed for automated synchronous circulation of data [10], [11].

Blockchain technology and smart contracts are one possible approach to creating a decentralized IT infrastructure for the Smart Grid. In the past few years, there have been studies applying blockchains to various elements of the Smart Grid ecosystem. However, the capability and performance of these techniques is a debatable question. The literature on smart contracts in energy markets - and particularly their interaction with the technical infrastructure of the Smart Grid - is limited and scattered. There is thus a need to consolidate these studies into a comprehensive understanding of the state of the art in smart contract design for the Smart Grid. Therefore, in the first part of this thesis, we present a systematic review of the performance, design, and limitations of blockchain-based smart contract platforms in electric power

systems. In this contribution, we focused on the research studies that both designed and implemented blockchain-based smart contracts for energy systems. We found that this literature falls into four categories: market operations, ancillary services, auditing and monitoring, and cybersecurity. We also investigate blockchain's data storage and scalability problems, and present possible solutions to these drawbacks.

Blockchain technologies are furthermore one possible avenue for increasing the resilience of the Smart Grid, by decentralizing the monitoring and control of system-level objectives such as voltage stability protection. They furthermore offer benefits in data privacy, as blockchains are cryptographically secured. In the past few years, there have been studies applying blockchains to various elements of the Smart Grid ecosystem. Some previous studies have investigated using smart contracts to provide ancillary services by tracking and managing energy distribution in the network [12] or organizing some Distributed Energy Resources (DERs) to act as voltage regulators and curtail their individual power outputs [13]. A few studies also focused on scheduling and trading energy among Energy Storage Units (ESUs) to minimize grid fluctuations [14] [15]. However, the performance of blockchain-based systems in real-time grid monitoring and control has never been empirically tested. Therefore as the second part of this thesis, we design such a system for distributed voltage stability control as a real-time control system for complex objectives, using PMUs as the data sources. We have implemented an existing distributed voltage-stabilty control algorithm (DVS [16]) as a collection of smart contracts operating on a permissioned blockchain. This is in contrast to the original implementation, which was built on the Distributed Coordination Blocks (DCBlocks) [17] framework, which is a classic distributed-computing platform. The DVS algorithm splits the power grid into multiple groups and principally operates on localized voltage measurements but must still ensure system-level stability.

In addition, we propose a new transaction processing model based on the sharding technique to tackle the performance and scalability requirements of the Smart Grid.

In this model, by growing the Smart Grid network, one or multiple local controllers can be assigned to different shards within the network, and a committee is selected for each shard. This allows many more transactions to be processed in parallel at the same time, further improving the performance of the system.

We used the Hyperledger Fabric platform [18] to implement and evaluate our proposed model. We implemented the Power flow simulation in the Matpower package [1] [19] in Matlab. We analyzed the performance of our implementation using the Hyperledger Caliper benchmarking tools [20], showing that our solution scales linearly with the number of shards compared to an un-sharded approach. To the best of our knowledge, this is the first study that presents a practical, scalable decentralized voltage stability algorithm based on blockchain technology.

1.3 Thesis Outline

The rest of this thesis is organized as follows. We provide a background for Blockchain and Smart Contracts, Sharding, and the Smart Grid in Chapter 2. Chapter 3 presents a literature review of smart contract design in blockchain-based energy systems. Chapter 4 outlines our scalable blockchain-based smart contract model for decentralized voltage stability using the sharding technique, and evaluation methodology and results. Finally, Chapter 5 concludes this thesis and proposes some potential future work in this area.

[1] https://matpower.org/

Chapter 2

Background

2.1 Blockchain

Some unknown person or persons using the pseudonym "Satoshi Nakamoto" introduced blockchains in 2008 when they proposed the first cryptocurrency, Bitcoin [21]. Blockchain is a type of Distributed Ledger Technology (DLT), which is a specific kind of database stored in a distributed fashion and shared among a set of nodes or participants. The data is recorded on the blockchain as a group of transactions called blocks. Each block recursively links to the previous block by referencing the cryptographic hash value of the previous block. To attach a new valid block to the ledger, all or some participant nodes must reach a consensus on whether the information is valid or not. Several consensus mechanisms have been introduced for blockchain platforms which provide a trade-off between performance and scalability. In addition, some blockchain platforms also employs digital signatures using an encryption algorithm to ensure that only authorized parties can read the messages. Combining all these techniques makes blockchain a secure, transparent, and tamper-resistant platform for various applications such as digital currency transactions, energy systems, Smart Grids, IoT, supply chain, etc.

Classical database systems maintain integrity by enforcing the ACID (Atomicity, Consistency, Isolation, Durability) properties for every transaction. Atomicity is the principle that either all elements of a transaction occur, or none of them do.

Consistency is the requirement that all outcomes of a transaction must satisfy all existing constraints within the database. Isolation refers to enforcing serialization for transactions that attempt to modify one or more common data items. Durability is the property that a transaction, which has been committed to the database, will be completed and written even in the face of disruptions (e.g. power failure while the transaction is still in an HDD buffer). [22] The ACID properties are relatively easy to enforce in single systems, but more difficult in distributed systems. An alternative set of properties, known as BASE (Basically Available, Soft state, Eventual consistency) [23], has been proposed for the distributed environment; however, blockchains do not fully support these properties. Tai et al. [24] suggested a new set of properties for blockchains called "SALT," (Sequential, Agreed, Ledgered, and Tamper-resistant). The Sequential property is that all blockchain transactions must be processed in sequential order. The reason is that, in consensus mechanisms, validating the transaction depends on the previous one, and transactions must be committed to the system in order. Agreed represents that the majority of the nodes in the network should agree on the validity of a transaction. Ledgered means that once a transaction is committed to the blockchain, no one can revoke or delete it. Tamper-resident indicates once a transaction is committed to the blockchain, it is impossible to alter it.

Many blockchain platforms have been introduced, which differ in various features to meet different system requirements. Some of these features include network accessibility (public or permissioned), the smart contract programming language, development complexity, performance, privacy, and cost. The first generation of blockchains supported a simple scripting language for programming transactions. But later, a new generation of blockchain platforms introduced the smart contract. Smart contracts have provided the foundation for developing more diverse, flexible and complicated blockchain-based decentralized applications. Nowadays, new blockchain platforms support general-purpose programming languages such as Golang, Python, Javascript,

and Java for implementing complex transactions.

In terms of network accessibility, there are two types of blockchain networks: public and permissioned. In public blockchains, everyone with an anonymous identity can join the network, submit a transaction, query data from the blockchain, or participate as a validator in consensus algorithms. Hence by increasing the number of blocks, public blockchains require a more complex consensus mechanism to maintain a distributed ledger at a large scale. Bitcoin and Ethereum [25] are the most popular examples of public blockchain networks. On the other hand, in permissioned blockchains, only authenticated users can join the network, and users could have limited access levels to the different parts of the network. Due to this initial filtering, permissioned blockchains can employ lighter consensus mechanisms. Hyperledger Fabric [18] and the Corda network [26] are examples of a permissioned blockchain.

2.2 Smart Contract

Nick Szabo [27] introduced smart contracts for the decentralized ledger in 1994. A smart contract is a self-executing or digital contract stored and run on the blockchain to enforce an agreement between parties involved in the transaction. Nowadays, by developing blockchain platforms that support high-level programming languages for smart contracts, we can develop more complex and flexible procedures in various business areas, such as digital asset exchange, supply chains, crowdfunding, and intellectual property.

Each smart contract can include multiple transactions, consisting of the code for a set of functions and the smart contract's initial state. They are typically used to automate an agreement's execution so that all participants can be immediately certain of the outcome without any intervention of trusted authorities or time loss. They can also automate a workflow, triggering the next action based on specific conditions. The smart contract code's location is dependent on the platform. For example, they can be stored on the ledger like transactions or installed on network

peers. Some blockchain platforms like Hyperledger Fabric, Tendermint, etc., support general programming languages such as Java, C++, NodeJS, Python, and Go for smart contracts. On the other hand, other platforms, such as Ethereum, use a bespoke language for implementing smart contracts. Ethereum utilizes the domain-specific language (DSL) Solidity [28] to write smart contracts, which is a Turing complete language that runs on the Ethereum Virtual Machine (EVM).

Smart contracts can be responsible for trading large amounts of money, digital assets, stocks, or data based on the application requirements. Although the consensus protocol ensures the faithful execution of smart contracts, it still has many security issues. Furthermore, different platforms and languages can cause smart contracts to exhibit distinguishing security vulnerabilities. Therefore, considering the security of Smart contracts is essential; for example, a small bug can lead to critical problems, such as a significant amount of money loss or privacy leakage. Previous studies show that these security issues are especially severe in public blockchains like Ethereum. In June 2016, an Ethereum Smart contract code bug resulted in a USD 60 million loss [29]. The attacker exploited the reentrancy vulnerability to recursively call the function and transfer Ether (Ethereum cryptocurrency) to his/her account. Therefore, to address these vulnerabilities, several techniques were suggested to improve smart contract security [30].

2.3 Consensus Mechanism

The consensus mechanism defines the process of validating transactions in blocks and reaching an agreement between all peers. The available and proposed consensus mechanisms differ in computation power, performance, scalability, and tolerance of disruptive behaviours. Nguyen and Kim [31] presented a survey on consensus mechanisms and classified them into two main categories: proof-based, in which one or multiple leaders are responsible for validating and attaching blocks to the ledger; and voting-based, in which numerous nodes vote for every block validation and, based on

consensus policy, a minimum number of positive votes is required for block validation. The primary difference between these two techniques is how they can choose the leader. The most common consensus mechanisms include Proof of Work, Proof of Stake, Delegated Proof of Stake, Proof of Importance, Raft, and Practical Byzantine Fault Tolerance.

Proof of Work (PoW) [32] was first published in 1993 by Cynthia Dwork and Moni Naor and was later applied by S. Nakamoto to Bitcoin and then later applied to Ethereum. The Proof of Work consensus algorithm is an incentive-based method in which each involved node solves a computationally challenging puzzle to create new blocks and gain rewards. The process is like constantly guessing until they solve the puzzle and find a value called the nonce. When a node finally finds the right solution, it is considered the winner of the current block and broadcasts it to the whole network simultaneously for verification and receiving a cryptocurrency prize (the reward) provided by the PoW protocol. Then nodes check the proposed block against cheating, and the collection of ordered transactions will be committed as a new block to the blockchain. PoW is resilient against tampering but requires very high computation and energy power (the Bitcoin network famously consumes more power than some European countries) and has trouble scaling to accommodate large numbers of transactions.

Proof of stake (PoS) was introduced by Peercoin 15 [1] to overcome and reduce the cost of PoW. Based on this method, the network chooses a miner based on the number of coins they are staking. In this case, the term staking refers to the act of validators committing funds to the system, by locking out their access to those coins. It means wealthier nodes would have a better opportunity to get selected as a miner on each iteration. The miner gets rewarded by proposing a correct block after other nodes validate it. PoS can provide increased levels of scalability, energy efficiency, decentralization, and security, but there is an equity concern that only nodes already

[1] https://peercoin.net/

having large coin holdings are able to effectively participate. This may disincentivize new miners from participating in the network.

Delegated Proof of Stake (DPoS) [33] introduced by Daniel Larimer, in 2014. This mechanism is similar to PoS but is faster and more efficient because the number of validator nodes is lower. DPoS presents a novel voting system in which "stakeholders" elect a subset of nodes participating in the block generation process. The voting power of stakeholders is proportional to the number of coins that each node holds.

Proof of Importance (PoI) [34] has been introduced by the NEM blockchain platform. The node or accounts with higher importance have a higher chance of appending a new block to the ledger. Account importance is measured by the frequency of blockchain usage and coin transfers. The network assigns a rating to each account according to its importance using graph theory methods.

Raft [35] is a Crash Fault Tolerant (CFT) consensus mechanism used by the Quorum, R3 Corda, and Hyperledger Fabric platforms. Raft states that each node in a replicated state machine can stay in any of the three states: follower, leader, and candidate. The leader is responsible for accepting client requests and managing log replication. The followers are passive and replicate the leader's decisions. When there's no leader in a Raft cluster, any follower can become a Candidate and vote for the next leader. The minimum number of votes required to operate a Raft cluster is $(\frac{N}{2} + 1)$, where N is the number of total members in the group. The candidate that receives the maximum votes will be selected as the next leader; otherwise, it reverts to the follower state.

Practical Byzantine Fault Tolerant (PBFT) [36] is based on the Byzantine Fault Tolerant consensus. Nodes in the PBFT model are ordered, with one node being the primary node (leader) and the others being the backup nodes. All of the nodes within the system communicate with each other to agree on the system's state through a majority, and also need to verify that each message was not modified during transmission. The algorithm effectively provides both liveness and safety as long as at most

$\frac{(n-1)}{3}$, where n represents total nodes, are malicious or faulty at the same time. The leader is frequently replaced in a round-robin type format or other defined protocol. Each consensus round includes three phases: pre-prepared phase, in which the leader proposes a new block as a proposal to the remaining nodes; prepared phase, in which other nodes broadcast their votes to the leader and other nodes; and the commit phase, which is for adding a new block to the blockchain. Istanbul Byzantine Fault Tolerance (IBFT) [37] is simulated by PBFT with some modifications. In contrast to PBFT, the set of validators in IBFT is not static, and it has a dynamic validator set which validators can be added to or removed from. Besides, there is no client to send the proposed block, and every validator can offer a suggesting block.

Once validated, blockchain transactions are applied to all the ledger's copies. As a result, we reach distributed trust among a set of untrusted peers, and we can execute trusted distributed applications. However, techniques and methods required for security and maintaining trust in the network are different in public and permissioned blockchains. In public blockchains, all participants are entirely untrusted; therefore, complex consensus mechanisms with high computational costs, such as PoW, are required. In contrast, in permissioned blockchains, due to the initial filtering, the involved nodes are semi-trusted, and so a lighter consensus protocol, such as traditional Byzantine-Fault Tolerance (BFT), is enough to achieve security and trust.

In permissioned blockchains, it is also possible to separate the trust model of transaction validation from the consensus protocol. The trust model and assumptions can be customized based on the application requirements by adapting the smart contract. Endorsement policies in Hyperledger Fabric basically allow users to define policies around the execution of smart contracts. These endorsement policies define which peers need to agree on the results of a transaction before it can be added to the ledger. Based on the required trust level in a smart contract and the involved parties, the number of peers required to endorse (confirm) transactions can be tailored [18].

2.4 HyperLedger Fabric

HyperLedger Fabric [18] is an open-source enterprise-grade permissioned DLT platform established under the Linux Foundation. Fabric was the first blockchain platform to support general-purpose programming languages for smart contracts, making it popular among developers with different programming skills. Figure 2.1 illustrates the architecture of the Fabric network. The architecture's major components such as peer nodes, clients, ordering service, Membership Service Provider (MSP) provider, channels, and Chaincode (Fabric's implementation of smart contracts). In the following, I will provide a review of each component of HyperLedger Fabric and describe transaction flow in this network.

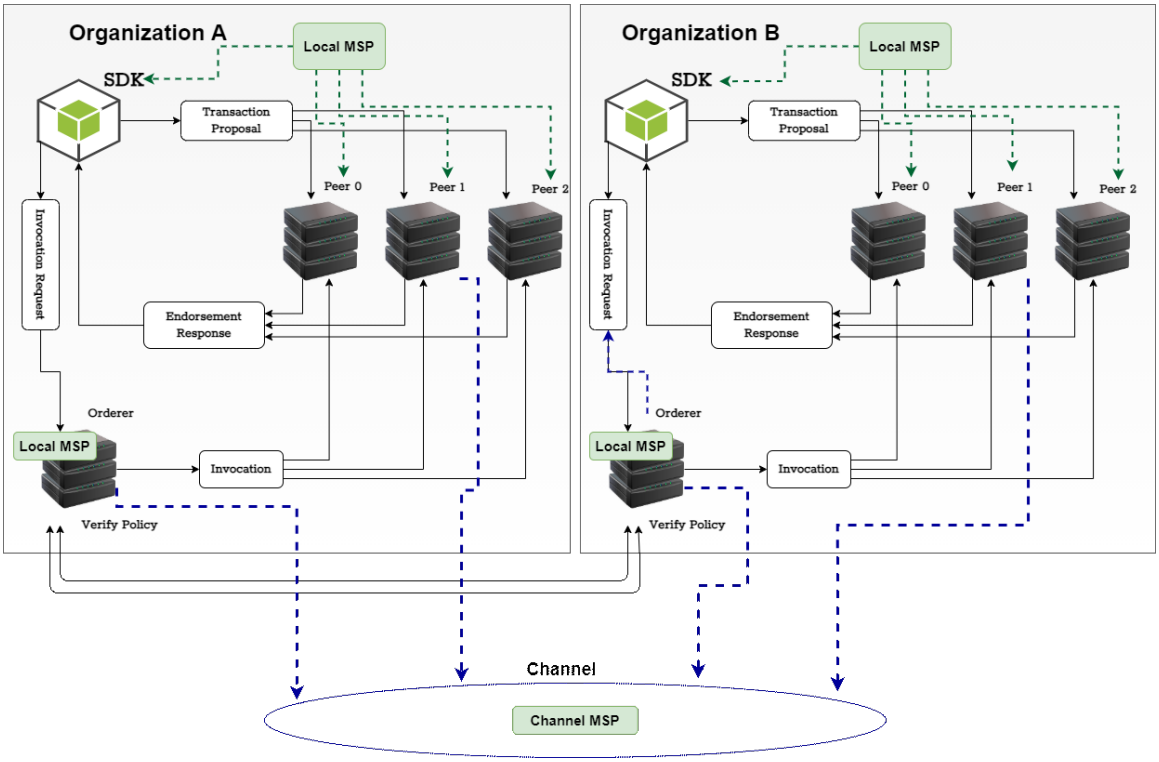

Figure 2.1: Architecture of HyperLedger Fabric Network

Client

Clients are applications or devices that act on behalf of a person or organization to propose transactions on the network. The client communicates with the network through a Fabric SDK in order to read or write the data in a ledger. HyperLedger Fabric offers several SDKs to support developing applications in various programming languages such as Node.js and Java.

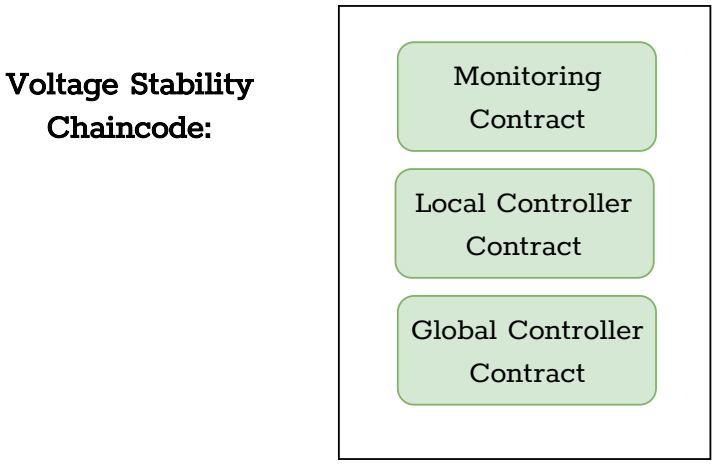

Figure 2.2: Chaincode Overview.

Chaincode

HyperLedger Fabric introduces chaincode, a component that implements application logic and runs during the execution phase. A smart contract represents the transaction logic that controls the life-cycle of a domain-specific program. A chaincode is a container of multiple related smart contracts for installation and instantiating. When the chaincode is deployed in the Fabric network, all smart contracts within the chaincode are made available to the application. Hyperledger Fabric users often use the terms smart contract and chaincode interchangeably. Figure 2.2 shows the overview of chaincode in Fabric.

Every chaincode has an endorsement policy attached, which applies to every smart contract defined within it. Before transactions generated by the smart contract can be identified as valid, the endorsement policy specifies which organizations must approve

or sign those transactions. For example, suppose an endorsement policy specifies that more than one organization must sign a transaction. In that case, the smart contract must be executed by a sufficiently large set of organizations to generate a valid transaction. In other blockchain platforms such as Ethereum or Bitcoin, any node in the network can submit valid transactions. However, endorsement policies in HyperLedger Fabric model the real world, in which trusted organizations in a network must validate transactions.

HyperLedger Fabric Ledger

A ledger stores states, which means recording the current state of an object and the facts about the history of transactions that led to the current state. Ledger states are, by default, expressed as key-value pairs. In Hyperledger Fabric, a ledger consists of two distinct parts: a world state, which is a database that holds current values of ledger states, and a blockchain state, which is a transaction log that records all the changes that have resulted in the current the world state. The blockchain data structure is very different from the world state because once written, it cannot be modified: it is immutable.

Orderer

Transactions in a blockchain network have to be written to the shared ledger in a consistent order. The order of transactions helps to ensure that the updates to the world state are valid when they are committed to the network. In most public blockchains such as Bitcoin and Ethereum, any node can participate in the consensus process; therefore, the system guarantees ledger consistency to a high degree of probability. In contrast, HyperLedger Fabric relies on deterministic consensus algorithms, which means that any block validated by the peer is guaranteed to be final and correct. The Fabric network has an ordering service that includes multiple orderer nodes to do this transaction ordering and allows the organizations to choose the ordering mechanism that best suits that network. HyperLedger Fabric provides four ordering

mechanisms: Raft, SOLO, Kafka, and Simplified Byzantine Fault Tolerance (SBFT). SOLO involves a single ordering node, and the transactions are ordered in chronological order to form a block. The Kafka mechanism is similar to Raft-based ordering and provides a crash fault-tolerant solution to ordering service. SBFT is both crash fault-tolerant and byzantine fault-tolerant, meaning that it can reach agreement even in the presence of malicious or faulty nodes.

Peer

A Peer is a node that maintains the state and a copy of the ledger and runs chaincode containers to perform read, query and write operations to the ledger. A peer receives ordered state updates in the form of blocks from the ordering service and maintains the ledger state. The four types of peer in the HyperLedger fabric include: Endorsing peer, Committing peer, Anchor peer, and Leading peer. Committing peers are responsible for committing the block received from the Ordering service in their copy of the ledger. Committing peers validate each transaction in the block, mark it as valid or invalid, and commit it to the block. All transactions, either valid or invalid, are all committed to the blockchain for future audit purposes.

Endorsing peers are a special type of committing peers which have an additional role in endorsing a transaction. They provide an endorsement of the proposed ledger update to the application but do not apply the proposed update to their copy of the ledger. The primary purpose of an Endorser is to simulate the transaction. The transaction is executed based on the smart contract on a private copy of the ledger of a peer and is not committed to the ledger during simulation.

Anchor peers are responsible for communicating across an organization as the Fabric network can extend across multiple organizations. These special peers are only authorized to discover and communicate with other peers on a channel. Each Member or organization on a channel may have multiple anchor peers to prevent a single point of failure.

Leading peers are responsible for communicating or disseminating messages from

the Ordering service to other peers in the same organization. These peers use the Gossip protocol to make sure that every peer receives the message, but they cannot communicate across an organization. If any Leading peer is not responding or is out of network, other available peers choose another leading peer, either by voting or at random.

Channel

A channel is a private "subnet" of communication between two or more specific network members to utilize the same network for data isolation and confidentiality. Members of channels are peers that are authenticated and authorized to transact on that channel. Each peer can be part of multiple channels and maintain multiple ledgers. Only members of a channel are allowed to see the transactions created by any member in a channel and are involved in consensus, while other network members do not see the transactions on the channel.

Membership Service Provider (MSP) and Identity

In Fabric, every participant or actor (clients) has some digital identity in the form of an X.509 certificate. This identity is used to verify each step of a transaction to check if it is from a valid source. Therefore, each actor requires a Public Key Infrastructure (PKI), which is comprised of a Certificate Authority (CA) and possibly even an Intermediate Certificate Authority (ICA) to issue an identity. This identity also includes attributes indicating the actor's permission level to access the different information and components in the Fabric network.

Fabric CA is a default certificate authority that issues identities by generating a public and private key. Fabric MSP verifies the identities issued by Fabric CA without revealing the actor's private key by maintaining a list of permissioned identities and ensuring that all nodes, especially all peers, recognize the same identities and authentications as valid. Figure 2.3 presents an overview of the MSP providers.

In Fabric Networks, nodes belong to an organization. An MSP defines the organization's administrators (admins), and the admin of the organization defines each

node's admin. MSPs occur in two domains in a blockchain network: locally on an actor's node (local MSP) and in channel configuration (channel MSP). Local MSPs are defined for every node (peer, orderer, etc.) and represent the permissions the node or who has administrative or participatory rights at that level. For instance, the local MSP on the peer allows for authenticating member messages outside or inside the context of a channel and defining the permissions over a particular node, e.g. who can install chaincode on a peer. Whereas local MSPs are represented as a folder structure on the file system, channel MSPs are described in a channel configuration and define administrative and participatory rights at the channel level. Channel MSPs are shared between channel members (peers and ordering nodes) to authenticate the channel participants correctly.

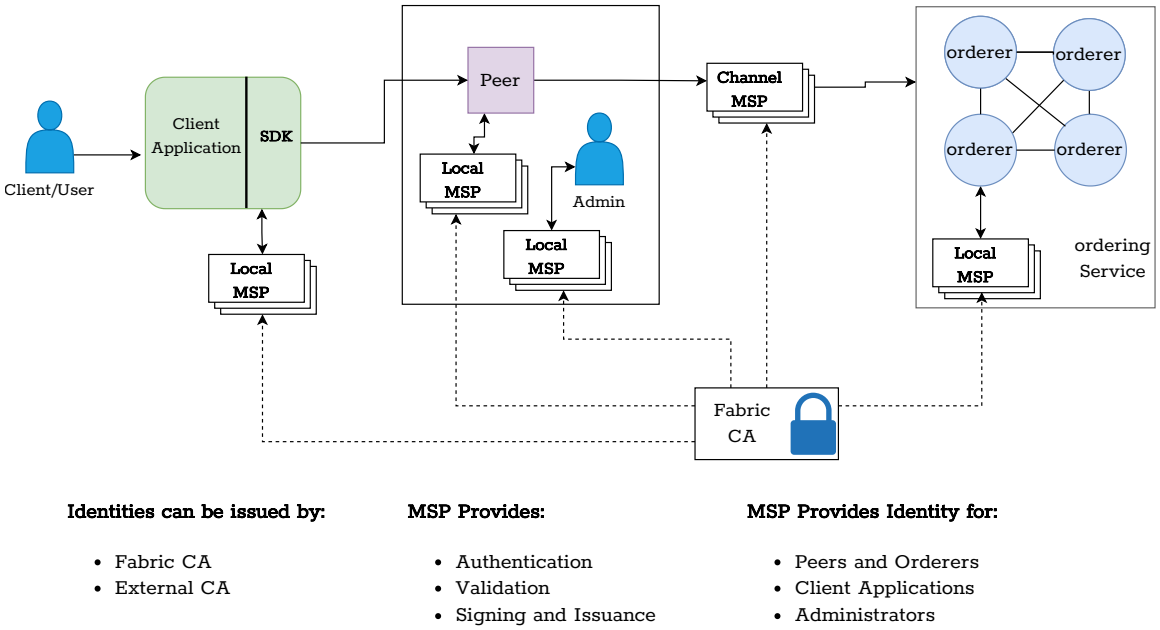

Figure 2.3: Membership Service Provider (MSP) - Overview

Transaction flow in HyperLedger Fabric

The first step in a transaction between two clients is setting up a channel and registering users with the Certificate Authority to authenticate to the network. Supposed Client A wants to purchase an asset from Client B. The chaincode that contains logic representing a set of transaction instructions for this purchase should be installed on

the peers and deployed to the channel, and an endorsement policy for the chaincode defined. When Client A initiates a transaction to purchase an asset, the request goes to peer A and peer B, representing each client. The Fabric SDK generates a transaction proposal, which is a request to invoke a chaincode function with specific input parameters to read or update the ledger. It then packages the transaction proposal with a unique signature by using the user's cryptographic credentials. After that, the endorsing peers verify that the transaction proposal is well-formed, the signature is valid, the submitter is authorized correctly to satisfy channels policy, and it has not been submitted already in the past (to prevent a replay attack). The endorsing peers use the transaction proposal inputs to invoke the chaincode, producing transaction results including a response value, read set, and write set. The ledger is not updated at this time, but the results and the endorsing peer's signature are passed to the SDK as a proposal response for verification. If the chaincode submits the transaction to the ordering service to update the ledger, the SDK verifies if peers endorsed the transaction. After confirmation, the SDK signs and sends the transaction, including the read/write sets, the endorsing peers' signatures and the Channel ID, to the ordering service. The ordering service receives transactions from all channels in the network, then orders transactions chronologically by channel and creates blocks of transactions per channel. The orderer is responsible for verifying transactions based on cryptographic signatures, policies, etc.; all orderers in each client-side must agree and sync to confirm the transaction. If the transaction is invalid, the orderer stores it in blockchain but does not update the ledger's state. If it is valid, the orderer sends the invocation to all peers in the channel to apply the operation. Each peer appends the block to the channel's chain, and for each valid transaction, the write sets are committed to the current state database. Then it emits an event to notify the client SDK that the transaction (invocation) has been immutably appended to the chain and inform whether the transaction was validated or invalidated.

Comparison of Ethereum with HyperLedger Fabric

Table 2.1: HyperLedger Fabric vs. Ethereum.

Features	**Blockchain**	
	Ethereum	**HyperLedger Fabric**
Confidentiality	Public / Private Network	Private Network
Maintenance	Ethereum Developers	Linux Foundation
Consensus Mechanism	POW- Proof of Work Mechanism	Pluggable consensus mechanism
Participation	Anyone	Organizations with Certificate of Authorization
Cryptocurrency	Ether or Ethereum	None
Programming Language	Solidity	Golang, JavaScript, Java
Transaction Fee	Yes (Gas)	None
Speed of Transaction	Less (15-20 Transaction per Second)	More
Security and Privacy	Less	More

Ethereum, like HyperLedger Fabric, is one of the most popular decentralized, open-source blockchains with smart contract functionality. A large number of blockchain applications have been developed for Ethereum; however, as a public blockchain, its use cases are different from Fabric. Table 2.1 briefly compares the different properties and functionalities of these two environments. Hyperledger can be very useful when an organization or a business wants to develop customized blockchain applications for business purposes due to the flexibility and capability of changing the whole underlying infrastructure. It also offers the facility to create a blockchain application maintaining the privacy of the organization's information. On the other hand, with Ethereum, anyone can join the network and create a node. Every such node possesses a copy of the entire blockchain. It is more useful for applications that do not need any confidentiality and can be developed and hosted by blockchain developer communities around the world.

2.5 Sharding

In large distributed public networks, like bitcoin and Ethereum, scalability has already been identified as a serious problem. The objective of increasing scalability in blockchain systems is to process a high number of transactions per second (throughput) without sacrificing security and decentralization. Besides throughput, several other factors, such as storage, cost, and latency, can also impact blockchain scalability. Storage issue refers to the problem of managing and handling the increasing demand of transactions recorded in the blockchain over time. Latency refers to the time between submitting a transaction to the blockchain and the first confirmation of acceptance by the blockchain. Latency will be increased as the number of transactions grows due to the peer-to-peer verification process. Once a transaction is confirmed, based on the blockchain type, the user may need to pay transaction fees to the miner; therefore, it will be much cheaper for the user to execute as many transactions as possible outside of the blockchain and then later record them as one transaction.

Sharding is one of several popular ways to boost the horizontal scalability in a blockchain, improve the throughput, and address latency problems. Shards are subchains built over the main blockchain that spread out the computational and storage workload across a peer-to-peer (P2P) network. In this case, each node isn't responsible for processing the entire network's transactional load, and only maintains information related to its shard. Each shard operates like a mini-blockchain with its own processing power and a dedicated set of nodes, which no longer need access to all the data on the main chain. Therefore the required computing power is further reduced, improving processing speed on every shard.

Transactions in sharding can be classified into non-cross-shard or cross-shard transactions [38]. Non-cross-shard refers to any transactions that only happen between nodes that belong to a single shard, and the transaction can only be verified by nodes in this specific shard. The security level of non-cross shard transactions de-

pends on the size of the shard due to the blockchain transaction consensus protocol. Cross-shard transactions occur between nodes from different shards, and the validating nodes are selected randomly from all shards. Furthermore, the type of sharding can be divided into transaction sharding, in which all shard nodes will store a complete ledger containing all verified transactions; and state sharding, in which each shard stores the verified transactions that its own nodes have processed. To avoid inconsistency in state sharding, all nodes that are related to the transaction keep the ledger consistent, including transaction nodes and validating nodes. For example, in cross-shard transactions, we might choose nodes every time randomly and use state sharding. Because the sharding system cannot determine which parts of nodes are responsible for storing the records, the blockchain system may lose state consistency [39].

Distributing transactions over a subset of nodes may cause higher security risks due to the smaller number of nodes that engage in transaction validating. Yu et al. [39] conducted a security analysis of the sharding in blockchain systems. They found that in both cross-shard transactions and non-cross-shard transactions, the security level can be affected by the rate of malicious nodes; therefore, most sharding designs use Verifiable Random Function (VRF) [40] to provide a random distribution method to assign validating nodes from the whole blockchain network. Moreover, some researchers have optimized the node distribution methods to improve the secure transaction rate, using techniques such as game-theoretic analysis [41] and trust-based shard distribution [42].

2.6 Smart Grid

Figure 2.4 illustrates an overview of the electric power system. Bulk generation is provided by large-scale electricity generators, comprising both conventional and renewable energy sources such as hydropower, gas, geothermal power, solar, and wind. Because electric power is commonly generated at a relatively low voltage like 30

kilovolts (kV), step-up transformers are used to increase the voltage and transfer the electric power to the high-voltage (e.g., 230/500 kV) transmission lines so that electricity can be dispatched at low losses. The electric power then reaches the distribution substations deployed at the load centers. There, the voltage is reduced by step-down transformers to a rather low level (e.g., 27.6 or 13.8 kV), and the electric power is distributed through the lower-voltage network. Pole-mounted transformers further reduce the voltage (e.g., down to 120/240 volt) such that residential customers can use it. In real-world applications, an additional sub-transmission system at a medium voltage level (e.g., 115 kV) can be set between the transmission and distribution systems to further reduce transmission losses [43].

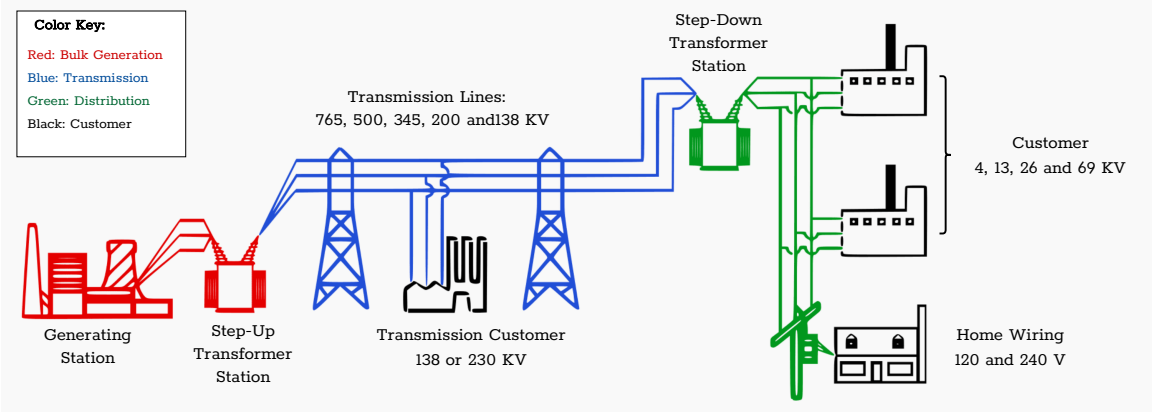

Figure 2.4: Overview of electric power system.

A grid is a network of electrical conductors that deliver electricity to certain points. The traditional grid can only transmit or distribute electric power. The Smart Grid, on the other hand, can store data, communicate, and make decisions, thus transforming the current grid into one that functions more cooperatively and responsively. It can also provide a platform that maximizes reliability, availability, efficiency, economic performance, and higher security from attack. The U.S. Electric Power Research Institute proposed the Smart Grid concept in 2001. Recently, it has been propelled again by promoting low-carbon economies in developing countries to satisfy the exponential increase in electricity demand while also reducing environmental degradation caused

by fossil fuel-based power generation.

The National Institute of Standards and Technology (NIST) presents standards and conceptual models that support planning, requirements development, documentation, and organization of interconnected networks and equipment composing the Smart Grid [44]. NIST focuses on seven key functionalities plus cybersecurity and network communications that are elements of the Smart Grid:

Demand response and consumer energy efficiency

Demand response is a mechanism that allows r encourages consumers in a grid such as utilities, business, industrial, and residential customers, to modify energy use during times of peak demand or when power reliability is at risk. Besides, by accessing each consumer's detailed energy consumption information, utilities can help each consumer save energy and optimize their energy usage behaviour.

Wide-area situational awareness

The goals of situational awareness are to monitor the behaviour of each grid's component in real-time to anticipate, prevent, or respond to problems before disruptions arise. For example, one of the grid's key challenges is maintaining voltage stability and ensuring that the system is reliable. Monitoring each power system component's behaviour or consumption helps sustain fixed tolerable voltage at every single bus of the network and prevents blackout.

Distributed Energy Resources (DER)

DERs are generators or electric storage systems interconnected with distribution systems. DER systems can include various generation and storage technologies such as renewable energy, combined heat and power generators (CHP), fixed battery storage, electric vehicles with bi-directional chargers, and controllable loads. DER systems can be used in local generation/storage, participate in capacity and ancillary service markets in the Smart Grid or be aggregated as virtual power plants. Based on several studies [45, 46], these functionalities help separate the local grid when power is disrupted and form a more adaptive resilient power system. But integrating DER

with a distribution system can also create challenges due to the uncertainty of most renewable sources such as solar panels, wind power, etc. For example, complicated optimization tools are needed to balance the network and control the supply-demand relationship. Otherwise, adverse events could occur such as a reverse power flow from the distribution system to the transmission system, creating unexpected congestion.

Energy Storage

Energy storage systems are regarded as promising solutions for providing ancillary services to electricity networks and playing an important role in developing Smart Grids. There are different ways to store energy, directly or indirectly. The pumped hydroelectric storage technology is one of the most common bulk energy storage technologies [44].

Electric transportation

Electric transportation refers to trains, trams, cars, buses and bikes which run on electricity. The Smart Grid will be needed for large-scale integration of plug-in electric vehicles (PEVs). Electric transportation could significantly reduce dependence on oil, increase the use of renewable energy sources, provide electric energy storage to alleviate peak-load demands, and dramatically reduce the nation's carbon footprint.

Network communications

It is critical for the Smart Grid to implement communication networks, both wired and wireless, which define a common semantic framework for enabling effective communication and coordination across inter-system interfaces. Effective communication means that each of the systems in the grid can understand and respond to the data provided by the other systems, even if the system's internal workings are quite different.

Advanced metering infrastructure (AMI)

AMI provides real-time monitoring of power usage. These advanced metering networks are integrated equipment, communications, and information management systems that create a two-way network between advanced meters and utility business

systems, allowing the collection and distribution of information to customers and other parties, such as the competitive retail supplier or the utility itself.

Distribution grid management

The automation of distribution systems is increasingly critical to the efficient and reliable operation of the overall power system. Distribution grid management can increase reliability, reduce peak loads, increase efficiency of the distribution system, and interact with distributed renewable energy sources.

Cybersecurity

Cybersecurity includes measures to ensure the confidentiality, integrity, and availability of electronic information communication systems and control systems, which are necessary for managing, operating and protecting the Smart Grid's energy, information technology, and telecommunications infrastructures.

Chapter 3

A Systematic Review of Blockchain-based Smart Contracts in Distributed Energy Systems

In this chapter we conduct a systematic review of smart contract applications in the Smart Grid. We first focus on blockchain-based smart contract design and implementation in different studies, and categorize models and a variety of use cases related to energy companies' operations and Smart Grid functionality. Finally, we investigate the storage limitations and alternative solutions for blockchain technology.

3.1 Introduction

Energy systems are undergoing tremendous change to adapt to the high penetration of renewable energy, increased energy demand and technological advancements. The Smart Grid, which leverages Information Technology (IT) to improve energy system efficiencies, is a key response to these needs. Blockchain technologies are one possible avenue for increasing the resilience of the Smart Grid by decentralizing the monitoring and control of system-level objectives and fulfilling contracts in diverse markets. The decentralized nature of the blockchain may be particularly important for the energy market as the penetration of residential prosumers offering microgeneration to the grid grows.

Using blockchain for distributed management of energy systems offers several ad-

vantages. The system is more reliable as data are protected against loss, tampering, and a single point of failure. Some blockchains allow parties that do not share a trust relationship to interact without the need for a third party guarantor [47]. The security and privacy of data can be protected through smart contracts' automated rules and the inherent features of blockchains [48, 49] while also providing cost savings [10]. Some blockchain designs can also promote social cohesion and a sense of community, preference satisfaction, and uncertainty reduction [50].

Smart contracts are the key component in designing and developing distributed applications based on blockchains, specifying how data is written to the ledger and analyzing stored data from the blockchain. In particular, for decentralized energy systems, smart contracts implement controllable procedures to verify and prepare energy data to be stored on the ledger, and process and analyze the stored data. Therefore, this review focuses on smart contracts' design and development patterns in decentralized energy applications. Previously, multiple literature reviews focused on distributed energy systems which either present a comprehensive systematic review [51] or investigate a particular aspect, such as cybersecurity [52], data analysis [53], or application areas of blockchain in the Smart Grid [54]. Table 3.1 summarized some of these review papers. However, to the best knowledge of the authors, there has not been a review that focuses on smart contracts within energy systems, particularly those smart contract designs that have been implemented as prototypes.

In this contribution, we present a comprehensive, systematic survey on a smart contract's performance characteristics and design that interacts in complex ways with the underlying blockchain network. It is thus essential to empirically test a smart contract's performance; without such an evaluation, the suitability of the smart contract for its intended application simply cannot be determined. Our research questions are thus as follows:

(Q1) *For what purposes are smart contracts deployed in energy systems, and how are smart contracts implemented?*

Table 3.1: Summary of recent reviews on smart energy systems

Paper	**Research Focus**	Smart Contract
Alladi et al, Blockchain in Smart Grids: A Review on Different Use Cases [54]	Several commercial blockchain deployments in the Smart Grid and various challenges that must be overcome in order to integrate these two technologies were discussed.	They discussed the importance and application of smart contracts in the blockchain-based Smart Grid, such as security and energy trading, but they did not review studies in this area.
Zhang et al, Big data analytics in Smart Grids: a review [53]	Discussed big data analytics and corresponding applications in Smart Grids like benefits brought to the present power system and improved customer service as well as social welfare in the era of big data by dealing with massive amounts of data from energy networks, meteorological information systems, geographical information systems, and other sources.	-
Ali et al, State-of-the-Art Artificial Intelligence Techniques for Distributed Smart Grids: A Review [55]	Reviewed of the state-of-the-art artificial intelligence techniques to support various applications in a distributed Smart Grid. Identified some limitations of the AI techniques presented in the literature.	-
Kushch et al, A review of the applications of the Block-chain technology in smart devices and distributed renewable energy grids [56]	A critical review of the existing technology in the smart cities and Smart Grid paradigms from the security perspective.	They defined and discussed the application of smart contracts briefly
Andoni et al, Blockchain technology in the energy sector: A systematic review of challenges and opportunities [51]	Provided a systematic review of 140 blockchain research projects and initiatives undertaken by companies and research organizations in energy sector.	Include some researches or companies that use smart contracts.The potential application of smart contracts were discussed in billing and competition.
Bao et al, A Survey of Blockchain Applications in the Energy Sector [57]	comprehensive review of how blockchain technology has been deploed in P2P energy trading system.	Reviewd some paper that used smart contracts, but their focus is on application of blockchain
Abdella et al, Peer to peer distributed energy trading in Smart Grids: A survey [58]	Provided a comprehensive survey in P2P energy trading including demand response optimization models, power routing devices and power routing algorithms.	Discussed the importance and application of smart contracts
Mollah et al, Blockchain for future smart grid: A comprehensive survey [59]	provided a comprehensive survey on the application of blockchain in Smart Grid in security issues.	Discussed researches that used smart contract in microgrid application, but their focus is on application of blockchain in security, trading monitoring and control
Zhuang et al, Blockchain for cybersecurity in Smart Grid: A comprehensive survey [52]	presented a survey include the latest insights of ideas, architectures, and techniques of implementation based on blockchain's application in the Smart Grid for cybersecurity.	Discussed the application of smart contract in security

Table 3.2: Classification of studies reviewed in the paper

Scientific Database	**Total number of papers**	**Journal**	**Conference**	**Workshop**	**Preprint**	**Other**
IEEE	35	17	19	1	0	0
ACM	0	0	0	0	0	0
Springer	7	4	2	0	0	1
Elsevier	10	10	0	0	0	0
MDPI	2	2	0	0	0	0
Other Journal	2	2	0	0	0	0
Other Conferences	1	0	1	0	0	0
Total	58	35	22	1	0	1

(Q2) *What are the limitations of the blockchain platforms in (Q1), and how might these limitations be addressed?*

To answer these questions, the authors have searched the Google Scholar database (which, while not curated, seems to capture a wider selection of papers than other platforms, e.g. Scopus) using four sets of keywords, "Energy System Blockchain," "Energy System Smart Contract," "Smart Grid Blockchain, and "Smart Grid Smart Contract." Our inclusion criterion for these reports was that the paper must present the design of the smart contracts and the implementation of the proposed system. Therefore, the studies that only mentioned the smart contract concept but did not address smart contract design elements in the context of the specific application are excluded from our study. Our initial pool of papers included 62 papers; however, after examining the papers thoroughly, we excluded three papers because the studies were based on substantially inaccurate assumptions concerning blockchain fundamentals and smart contract objectives. Table 3.2 summarizes the sources and publication types of the selected primary reports. To address (Q2), after thoroughly reviewing the papers for the first research question, we identified two common limitations of blockchain-based solutions for energy system applications: on-chain data storage and interoperability.

Our first finding is that common patterns in the design and responsibilities of smart contracts, development constraints, platform customization, and case studies

Table 3.3: Electric power system planning and operation functions.

Function	Time frame	Smart contracts
System planning	1 – 10 years or longer	not applicable
System maintenance	1 week – 1 year	The potential can be investigated
Unit commitment	4 hours – 1 week	Ancillary systems, cyber security, auditing and monitoring
Economic dispatch	10 minutes – 4	Ancillary services
Regulation, control, and protection	10 minutes or shorter	Ancillary systems, cyber security, auditing and monitoring

in the power system lead to a subdivision of our primary sources into four groups. We thus categorize the studies as focusing on Trading and Market, Ancillary Services, Auditing and Monitoring, or Cybersecurity. Table 3.3 maps some of these categories to planning and operations in electric power systems [60].

The remainder of this chapter is organized as follows. In Section 3.2, we classify the selected literature into four main categories and identify the major research themes within each category. Section 3.3 introduces solutions and methodologies for data storage problems within blockchain applications, including both on- and off-chain data storage. Section 3.4 elaborates on interoperability between blockchains for increasing scalability and performance. Finally, we offer a summary and discussion of future work in Section 3.5.

3.2 Smart Contract Design in Energy System Case Studies

We categorized papers based on the application and design of smart contracts in the power grid system. In each following section, we elaborate and explain the problem and scope of each application and investigate the methodology and smart contract design that each paper proposed for each function. Table 3.4 presents a summary of energy system studies based on blockchain.

Table 3.4: Summary of energy system studies based on blockchain

Application	Summary
Market operations	Auction holdings [47–49, 61–73]
	Payment operations [47, 49, 61, 64, 71, 72, 74–77]
	Reduce energy cost and demand management [10, 70, 73, 78–81]
	Demand and Supply Optimization [65, 73]
	User registration and collect energy market data [73, 82–84]
	Energy Transfer[72, 85, 86]
Ancillary services	Supply and demand management [15, 83, 87]
	Charging coordination mechanism for energy storage units [14]
	Optimization and control of energy resources [74]
	Voltage regulation [12]
	Agreements for shared control of energy transfer processes [88]
	Proportional Fairness control strategy to avoid power surplus [13]
	Electronic Vehicle charging/discharging scheduling algorithm [89]
	Control approach for Battery Energy Storage System [90]
Auditing and monitoring	Monitoring and tracing energy consumption from the Smart Grid [83], [84], [91], [62]
	User Interaction and behavior evaluation [92]
	Data Monitoring and Sharing Mechanism [93]
Cybersecurity	Privacy-preserving approach to protect energy trading information [94]
	Edge model for a Smart Grid network [95]
	Keeping vulnerable smart meters out of the network [96]
	Atomicity of data transaction [75], [76]
	Temporarily blockchain network [97]
	Anonymous authentication and key agreement protocol for the edge-computing-based Smart Grid [98]
	Detection of disruptive behaviour of electrical power [91]
	The dynamic join-and-exit mechanism [99]
	Using Cryptography tool [100]
	Access control [48, 101]

3.2.1 Trading and Market

Trading energy can take different forms based on the types of parties in the grid. For example, in the energy market, two utility companies can sell power to each other (utility-to-utility), or one utility company sells the power to individuals or houses (utility-to-consumer). As the usage of roof-top photovoltaic (PV) panels and distributed wind-driven generators increases, traditional energy consumers are becoming prosumers (who can both consume and sell energy) [102], which establishes a (peer-to-peer) P2P energy market. P2P energy trading refers to direct energy trading between prosumers and consumers without conventional energy suppliers. Internet of Things (IoTs) and smart-meter technologies also revolutionize the energy market and enable real-time measurement of the power system [103]. A decentralized energy trading mechanism can take advantage of the above changes and facilitate the energy market. Distributed ledger technologies and smart contracts can allow a generating unit to trade with a consumer or power utilities directly via autonomous trading agents, cutting out the third parties [47] and reducing the cost of energy [80]. Smart contracts can search for the best deal in the marketplace that satisfies a consumer's forecast demand for a given time period, improving the efficiency of the operations and reducing the peak load [48]. Agreements between consumers and prosumers would be securely recorded in the blockchain and automatically executed at the scheduled time. Payments would occur automatically at the time of delivery as specified in the agreed contract. Furthermore, creating a global market for renewable resources and setting incentives for eco-friendly consumer behaviour can encourage investment in renewable generation plants [104].

Table 3.5 presents the studies we reviewed that designed blockchain-based smart contracts for the energy market. Figure 3.1 is a semantic diagram that classifies the smart contract design and usage in the energy market studies. As you can see, smart contracts are mostly used for registering and collecting data, auction implementation,

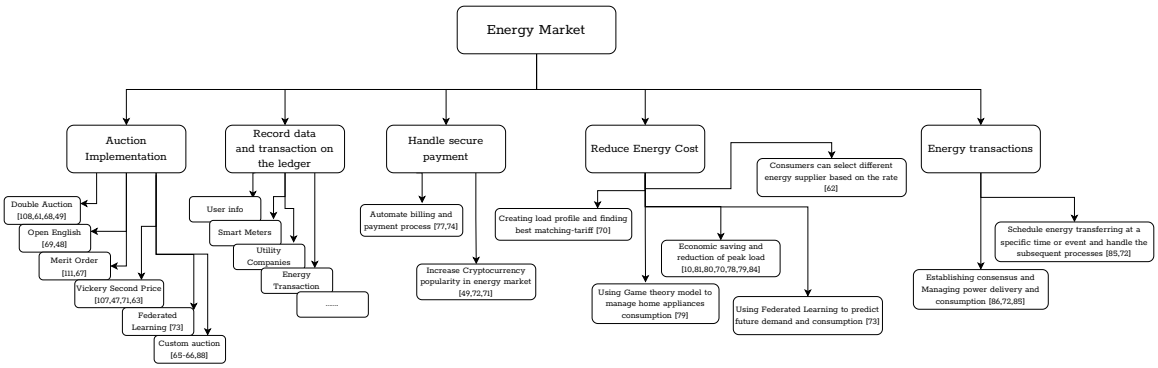

Figure 3.1: Semantic Diagram for Energy Market.

payment, transferring energy virtually, and reducing energy costs.

User Registration and Data Collection

The first task of smart contracts in the energy market is user registration and authorization. The market participants are required to be authenticated and authorized to interact with the blockchain, submit transactions, and query data from the ledger. Most of the studies in the energy market area use blockchain to store and process privacy-sensitive information of users [83]. Furthermore, user behaviour and activity can be tracked, allowing the utility to detect malicious behaviour [91]. Kang et al. [84] implemented user registration smart contracts, in which authorized consumers are also able to monitor energy usage and energy remaining in real-time. Prosumers are registered along with their conditions and price for selling. The contracts can also match consumers seeking to purchase energy with sellers.

Smart contracts collect the necessary data from the off-chain components and record them on the ledger for the forthcoming operations. These data can be collected from different and separated entities, such as smart meters, prosumers, utility companies, power generation centers and more. The collected data includes seller data (such as the quantity of available PV energy), bidder data (auction bids) and closed bid data [47, 49], the usage preference of each user [83], recording/updating utility company information, recording/updating demand information [82], and smart meter readings [77] [86].

Table 3.5: Energy trading market studies

Study	Blockchain platform	Description	Trading Mechanism	Implementation	Perf. evaluation
You 2019 [78]	-	proposed a demand response model & Reduce energy cost	-	-	-
Hahn 2017 [47]	Private Ethereum	Established a trustworthy market for prosumers	Vickrey second-price auction	Yes	-
Kumari 2020 [48]	Public Ethereum	Proposed a secure energy trading scheme called ET-DeaL	E-auction	Yes	Yes
Wang 2018 [10]	-	Using game theory model for demand-side management and creating an efficient trading system	-	-	-
Han 2020 [49]	Private Ethereum	P2P energy trading system	Double auction	Yes	Yes
Hu 2019 [87]	-	Trading mechanism for energy power supply and demand (EPSDN)	-	Simulation test	-
Mengelkamp 2018 [61]	Private Ethereum	Presented local energy market for 100 residential households	Double auction	Yes	-
Thomas 2019 [88]	Ethereum	Using smart contracts to make agreements for shared control of energy transfer processes.	Highest Combined Offer (HCO) and Ranked Preference Selection (RPS)	Yes	Yes
Afzal 2020[79]	Ethereum	A Distributed demand side management using game theoretic model	-	-	-
Khattak 2020 [105]	Hyperledger Fabric (Hyperledger composer)	Automate the bidding process based on supply and demand	-	Yes	-
Munsing 2017 [74]	Ethereum	Proposed a distributed optimization and control approach	-	simulation in SCE 55-bus test network	-
Amanbek 2018 [63]	-	Presented a novel method for a decentralized transactive energy management system	Modified Vickrey Second Price auction	Simulation test	-
Myung 2020 [65]	Ethereum	Using blockchain to reduce the wasted energy	Custom auction	yes	-
Nakayam 2019 [82]	Private Ethereum	Transactive energy market for solving an economic dispatch problem of Distributed Energy Resources.	-	Yes	-
Sabounchi 2017 [64]	Ethereum	-	Custom auction	Simulation: SunPower SPR-305E	-
Khalid 2020 [80]	Private Ethereum	A hybrid P2P energy trading market to reduce energy cost	Customized bidding	Yes	-
Mengelkamp 2018 [104]	Tendermint	An energy market to encourage investment in renewable generation plants and locally balancing supply and demand.	-	Yes	-
Dimobi 2020 [66]	Hyperledger Fabric	A peer to peer transactive energy operation within a microgrid	Auction-less with normalized sorting metric & a simple auction with penalties	Yes	-
Heck 2020 [67]	Private Ethereum	A local energy market	Merit order	Yes	-
Brousmiche 2018 [68]	Ethermint private	An agent-based simulation framework to implement a distributed energy market	Double auction	Yes	Yes
Monroe 2020 [50]	Power Ledger	An agent-based model for energy trading market	-	Multiagent Simulation and the Mason Library	-
Kang 2018 [84]	Private Ethereum	Renewable energy trading platform	-	Yes	-
Seven 2020 [69]	Public Ethereum	Virtual power plant trading	Open English	Yes	Yes
El-Syed 2020 [72]	Ethereum	-	Prosumers add offers and consumers select an offer they are interested in	Yes	-
Yang 2020 [15]	-	Proposed a novel Automated Demand Response framework	-	Simulation in CPLEX	-
Wen 2020 [106]	Hyperledger Fabric	-	-	Yes	-
Kounelis 2017 [77]	Ethereum	-	-	-	-
Lombardi 2018 [96]	-	Implemented energy trading and auctions transactions as well as security enhancement features	-	-	-
Wang 2020 [75]	-	A distributed Platform for managing access and sharing the data generated by smart meters and smart appliances	-	-	-
Dorri 2019 [76]	Private Ethereum	Participant can directly negotiate the energy price in a secure way	-	-	-
Dorri 2021 [97]	Hyperledger Fabric	proposed a blockchain network that records blockchain's transactions temporarily for the purpose of energy trading	-	Yes	Yes
Suther 2020 [62]	Ethereum-React	-	-	Yes	-
Sexana 2019 [70]	Hyperledger Fabric	An energy trading market mechanism for residential communities to reduce overall peak demand and electricity bills	-	Yes	-
Muzumdar 2021 [71]	Ethereum	Proposed a distributed trustworthy and incentivized trading platform	Vickrey auction	Yes	Yes
Zheng 2018 [100]	Ethereum	A trading system based on the combination of consortium blockchain, proof of stake consensus mechanisms and cryptography tools	-	-	-
Bouachir 2022[73]	-	Proposed a Federated Learning model based on blockchain for P2P energy market	-	Yes	Yes

Market mechanism

There is a need for distributed markets to enable energy exchange in transactive environments without the need for trusted third-party services between the buyer and seller entities. Smart contracts can execute the trading and payment rules without human interaction and enhance the security and fairness of energy trading. Multiple studies use smart contracts to implement auction mechanisms to form power purchase contracts in the energy trading market. These include double auctions, Open English auctions, Vickrey second-price auctions, and several other formats. In an auction market, buyers enter competitive bids, and sellers submit competitive offers simultaneously. The price at which a commodity trades represents the highest price that a buyer is willing to pay and the lowest price that a seller is willing to accept. Matching bids and offers are then paired together, and the orders are executed. The winners of the auction sell and receive power according to their bids, and the whole process then repeats for the next time period. For residential dynamic pricing schemes, the auction might run as often as every five minutes. Comparative studies of smart-contract-based auction models might involve examining market dynamics (e.g. market efficiency, comparing bidding strategies) [71] [49], or network characteristics in the underlying blockchain network (e.g transaction latency or throughput) [71]. Both involve simulating the auction model on a chosen blockchain network with various numbers and classes of participants. Based on the auction technique and implementation, the studies are categorized as follows:

A) Vickey Auction:

A Vickrey auction or sealed-bid second-price auction (SBSPA) [107] is a type of sealed-bid auction in which bidders submit their bids without knowing the bid of the other people in the auction. The highest bidder wins, but the price paid is the second-highest bid. Hahn et al. [47] use this technique to match consumers and prosumers. Vickrey auctions require sealed bids that protect parties from viewing other bidders, but blockchain transactions are public, and everyone can view other

transactions and use that information. Therefore, they implemented two different transactions of CommitBid and RevealBid. At first, all consumers commit their bid via the CommitBid function in the specified time. Then, the bids will be revealed, and the Vickery auction algorithm determines the auction winner and clearing price. Muzumdar et al. [71] implemented an energy injection contract that specifies energy available for the auction iteration and a bidding contract specifying total energy demand. A consumer with the highest bid wins the auction but pays an amount equal to the second-highest bidder's bid as an incentive. The auction is recursively carried out for the available energy and bidders without changing their bid value until all demands are satisfied, or available energy becomes zero. The loser of the auction can still receive energy at a government-regulated rate. The proposed framework satisfies more bids than the existing methods, such as the double price and the English auction method. They considered 20 consumers and 19 prosumers for testing energy trading with three different bidding strategies: Truthful bidding, Over-bidding, and Under-bidding. Based on different scenarios, they found out that the proposed framework forces consumers to do truthful bidding. Truthful bidders are auction winners and get a better deal than the feed-in tariff and get higher pay off than other bidders most of the time. Amanbek et al. [63] proposed a Modified Vickrey Second Price (MVSP) auction to solve the competition problem in transactive energy systems when multiple participants in the market have excess energy to sell. The energy trading is implemented in this auction based on locational marginal prices (LMPs), number of buyers, energy demand and energy availability. LMP refers to the price of electricity in real time at points across the regional high-voltage transmission system. This pricing is fundamental to competitive wholesale power market transactions. In MVSP auction prices are estimated by replacing bids with LMPs. Winners are decided based on lowest price estimate among sellers, whereas bid of buyers indicates maximum price at which they are willing to purchase the energy. In addition, the information on remaining energy demand is available at all times and encourage sellers to offer lower

prices when total demand is lower than supply.

B) Double Auction:

A double auction [108] is a methodology for buying and selling with multiple sellers and buyers. Buyers or consumer submit their bids, and potential sellers offer their ask prices to the market. Then it clears the market with a certain price. (A market clearing occurs when supply and demand are equal. The buyers and sellers use that price to exchange goods in the market.) Han et al. [49] proposed an energy trading mechanism based on a double auction consisting of closed bidding and energy exchange transactions implemented by smart contracts. In the closed bidding transaction, each producer and consumer can submit the bid amount, the bid price, and the energy type or preference. In the energy exchange transaction, the producers' offers are sorted in increasing order and the consumers' in decreasing order. Buyers and sellers are matched based on their energy type and price. They also evaluated the auction mechanism through a simulation running on the Ethereum platform consisting of nine players, including four consumers, four producers, and the distribution system operator (DSO). During a one-day simulation, energy suppliers and consumers submitted bids every thirty minutes, and then the system executed the energy exchange and settlement process. The evaluation result indicates that 83.72 percent of energy is successfully traded through simulated auctions, and the average clearing prices were always a number between the bid prices. The DSO balanced only 2.91 percent of the total actual energy because the mechanism is designed to incentivize prosumers to bid honestly. They compared their method with the conventional double auction methods [109, 110], and found that the proposed method is able to produce more accurate market quotations.

Mengelkamp et al. [61] implemented a closed double auction on a private blockchain. In each period time (t), each prosumer and consumer submit one bid or offer price to the market to satisfy trading demand in the next period (t+1). The lowest bid price that can still be served determines the market clearing price. Any surplus or deficit is

balanced by trading electricity with standard energy provider prices. Brousmiche et al. [68] implemented two functions via smart contract: proposeBid and proposeAsk to find a critical point where the demand volume meets the supply. Both take the volume of energy, the desired unitary price, and the target market turn. Additionally, the utility prices of energy (prices of the supplier) can be modified in real-time by authorized parties.

C) English Auction:

Seven et al. [69] implemented an auction contract adopting the "open English auction". First, each auction should be initialized by an auction owner (specifying starting price, duration, the bid increment amount, etc.). Each auction contract also should have a start and end time, but the auction owner may need to cancel the auction and withdraw the winning bid in unexpected situations. After the auction starts, any agent can submit a bid if no restriction rule is made and the auction has not been cancelled or ended. In auctions, users try to bid the maximum amount that passes the auction's highest bidder. If no competing bidder challenges the standing bid within the specified time by the owner, the standing bid becomes the winner. In the case that no bidder accepts the starting price, the owner either begins to lower the starting price in increments or bidders are allowed to bid prices lower than the starting price. When a customer submits a new bid greater than the previous one, the current highest bidding level is calculated as the previous top added to the bid increment amount, which the auction owner sets in the beginning. The fairness of the competition is secured in this way; otherwise, rich participating parties could overact easily to win all the auctions. Unlike other auctions, the English auction seems fully transparent and reveals the identity or the existence of all bidders and their bids. Kumari et al. [48] also established similar strategies for energy auction between prosumers and consumers called E-auction. They also assigned a time slot for the E-auction to handle the late response from users.

D) Merit Order:

Many energy markets use the merit order method to determine the market clearing price, such as The Alberta Electric System Operator (AESO) [111]. In the AESO market, suppliers enter offers, which are the pair of available capabilities and prices, seven days ahead of the delivery hour or any settlement interval. They are able to change the volumes any time and the price up to two hours before the settlement interval. For each hour of the day, the offers are sorted from the lowest-priced to the highest-priced. Then, the AESO's system controllers first dispatch the lowest-priced offers from the bottom and move up towards the higher-priced offers until meeting all electricity requirements and demand. The intersection of the power demand and power supply determines the clearing price or marginal price (SMP) for electricity. The SMP is calculated every minute and is used to calculate the hourly settlement price, also known as the Pool price. The Pool price is calculated as the average of all SMPs in each hour. Both SMP and Pool price reflect the market economies. In this method, Power stations with high generation costs such as gas or coal are vulnerable to being pushed out of the market as they cannot compete at lower prices from renewable installations such as wind turbines and photovoltaic installations. Heck et al. [67] implemented the merit order mechanism via smart contract in two phases: ask and bid. Ask includes an electricity amount, price and the electricity type that the prosumer sells. During each period, the smart contract reads generation and consumption data from participants' smart meters and creates an ask in case of excess. In case of consumption, the smart contract requests the latest smart meter reading and the participants' price preference to create a bid, including the amount and the preference based on the energy type.

E) AI method:

AI techniques in energy markets and P2P energy trading platforms can be used to predict energy production, demand and price to develop bidding strategies. Bouachir et al. [73] proposed a Federated Grid based on blockchains and smart contracts to automate energy trading between prosumers and consumers while providing trust and

privacy among all participants. They used various smart contracts to handle transactions such as registering prosumers and consumers, calculating prices, microgrid information, energy sharing, and federated learning. Federated learning contracts predict the future demand and production and select the various participants for the coming auction round. In the federated learning process [112], a machine learning model is distributed across multiple nodes, and the model is trained on the data of each node in a decentralized manner without sharing or transferring user data. Then all the parameters will be gathered and aggregated for use on a global model. Therefore, in the paper mentioned above, they implement two main functions with smart contracts. The first is GetModel, which allows the selected participant to gather the global learning model parameters from the contract. The second function, SetModel, allows sending or updating the parameters at the end of each round of the auction. The machine learning model enables the prosumers to decide on their participation and strategies in the energy market.

F) Other Settlement Mechanisms:

Some of the studies design a new method or similar auction methodologies to coordinate a microgrid market. Dimobi et al. [66] proposed two auction-less mechanisms and one simple auction scheme. In the auction-less method, a rate structure is predefined and categorized into three tariffs: a retail rate, which is the price that consumers pay to buy from the grid when there is excess demand; a community rate which is the price to exchange energy in the grid; and a wholesale rate which is the price for suppliers that sell power to the grid. A net value is calculated by subtracting consumption from production. The parties are sorted based on their net energy values, and the smart contract transparently updates their account balances. They normalized this schema by sorting the producers in ascending order and the consumers in descending order. They also proposed a simple auction mechanism that allows each prosumer to offer their price, and the community rate is the market clearing price determined by the auction mechanism. After matching the buyers and sellers, actual generation

and consumption are verified, and if there is a deviation from bids, penalties will be assessed. Sabounch et al. [64] implemented a simple auction in a smart contract. In each auction iteration, they received the currently available resources from the seller coalition and announced them to the buyers. They then wait until a specific time or until all buyers' bids are received to compare and announce the winner. Myung et al. [65] implemented an auction algorithm including 5 phases. The first phase is the initialization when the smart contract invokes auction data from the seller and initializes parameters such as the amount of power supply, time of supply, minimum bidding, and auction time. The next phase is Bidding, in which the buyers submit their bids based on the current auction. The bidder with the highest prices will be chosen after the time is up. Then all the buyers redeem their bidding amount at the withdrawal phase, and the seller transfers the agreed amount of power at the Supplier phase. Thomas et al. [88] used smart contracts to make agreements for shared control of energy transfer processes. They define two sets of algorithms for this purpose: The Highest Combined Offer (HCO) algorithm selects the highest combined bid from both network operators. The Ranked Preference Selection (RPS) algorithm selects the bid with the lowest summed rank of the options.

Secure Payment and Settlement

Once a power purchase contract is agreed upon, the next step is handling a secure payment between consumers and producers to pay for the purchased power. Considering the transparency, lack of third parties, and the existence of Cryptocurrencies, several types of research in the energy market used blockchain technologies and smart contracts to execute a secure payment [49] [64] [61]. The enhanced security features of blockchain help increase data privacy, identity management and resilience towards cyber-threats. In addition, some studies designed or used atomic protocols to ensure the atomicity of data transactions in computing results, release and payment [75][76]. For increasing the atomicity of transactions, several methods have been introduced. For example, we can block the money while transferring energy and release it after it

is transferred successfully to the consumers [76]. In other examples, both consumers and producers should pay a certain amount of money at the beginning of the auction. The blocked money will be released after the energy is transferred successfully and the prosumer receives their settlement.

If the blockchain is integrated with metering infrastructure, this provides a platform to track energy produced and consumed at each endpoint and inform consumers about the sources, whether it is renewable energy or comes from nonrenewable energy sources, and the cost of their energy supply [77]. Hence, it can provide the opportunity for automated billing and payment processes for consumers and distributed generators. After the smart contract receives the final schedule and clearing prices for exchanging power, a second smart contract can be designed to read the information from the trusted metering device and compares the result with the first smart contract. Then, it can compute penalties and automatically handle payments and charges based on that information [74].

An increasing number of utilities accept cryptocurrencies for energy and electricity payments. E.g. the Australian startup PowerLedger operates an energy trading network based on a P2P blockchain using the POWER token for payment [113]. Likewisely, many researchers in this area employ existing cryptocurrencies or create new tokens for payment in their distributed platform, including Han et al. [49], El-Syed et al. [72] and Muzumdar et al. [71].

Reducing Energy Cost

Implementing a blockchain-based energy trading market mechanism can reduce consumer electricity bills by eliminating transaction costs incurred by third parties [70]. It can also preferentially purchase energy from low-cost providers such as renewable energy sources in the grid. Furthermore, distributed energy management provides a platform for monitoring energy consumption and production in the grid [10] and finding the best match for each consumer based on their load profile and demand [81] , which can lead to economic savings, reduction of peak load [80], and enhanced

market efficiency. Knirsch et al. [81] proposed a framework based on the blockchain and smart contract to match load profiles from utility providers with load profile forecasts from customers in order to find a matching tariff without directly revealing any load profiles to any involved party. Their protocol finds the best-matching tariff for a customer with 93.5% accuracy while ensuring transparency, verifiability and reliability. Saxena Farag et al. [70] proposed a blockchain-based energy trading market mechanism for residential communities to reduce overall peak demand and electricity bills. They evaluated the proposed platform with two sets of data, which shows that the distributed platform can reduce peak demand by 46% and save 6% of total tariffs. Bouachir et al. [73] proposed a Federated Learning method to predict demand and consumption during each auction round. Their experimental results showed a 17.8% decrease in energy cost for consumers and a 76.4% decrease in load over utility grids.

Smart contracts are also used to implement a game theoretic model to handle and schedule the time and usage of home appliances to reduce individual costs [79]. This can help shift the operation time slot of each device from peak demand to off-peak demand, which reduces the need for additional peak generation and thus energy cost. Furthermore, due to market transparency it is also possible for smart contracts to change a user's energy demands in real-time in response to those of other users or changes in energy production (i.e. demand-response pricing) [78].

Kang et al. [84] designed an automated renewable energy market for residential consumers and prosumers using smart contracts. Using private blockchain associated with additional security layers reduces the time and cost of market process operations. Suther et al. [62] proposed a smart contract for local energy trading amongst residential users. Consumers have the authority to select from different energy suppliers. They can decide based on the rate, which leads to energy affordability through more competitive prices.

Virtual Energy Transfer

Smart contracts can potentially be used to schedule the transfer of energy between

consumers and producers. Multiple transactions can be defined to set rules, track the status of agreements or define schedules for delivering power. For example, smart contracts are used to predefined rules, such as delivering at a specific time or event and responding to particular situations [85]. In addition, each smart contract can invoke another transaction or function, which can help to handle subsequent processes after delivering the power [72]. Utz et al. [86] also established a consensus between various meter points by smart contracts, linking each meter point to a balancing group for settling energy delivery and consumption.

3.2.2 Ancillary Services

Ancillary services ensure that the interconnected electric system provides acceptable power quality for the current demand and supply, and ensure reliability and security. Based on the number of participants, and the size and complexity of the transmission system, Electric System Operators provide different services to ensure reliability when there is an unexpected imbalance in the grid. For example, The Alberta Electric System Operator (AESO) in Alberta, Canada [111], procures ancillary services including Operating Reserve (OR), Transmission Must-Run (TMR), Black Start, and Load Shed Services for imports (LSSi).

1. **Operating Reserve**: These reserves provide additional capacity or frequency support into the system, when the present energy supply is not adequate for the present demand. Regulating reserves are immediately available whenever a momentary imbalance in supply and demand would cause a voltage sag. Spinning reserve refers to generators that are synchronized to the grid but not yet delivering power; they can quickly inject substantial additional energy into the grid when needed. Supplemental ("standby") reserve refers to generators that are available but not yet synchronized to the grid.

2. **Transmission Must-Run**: TMR generation compensates for insufficient local

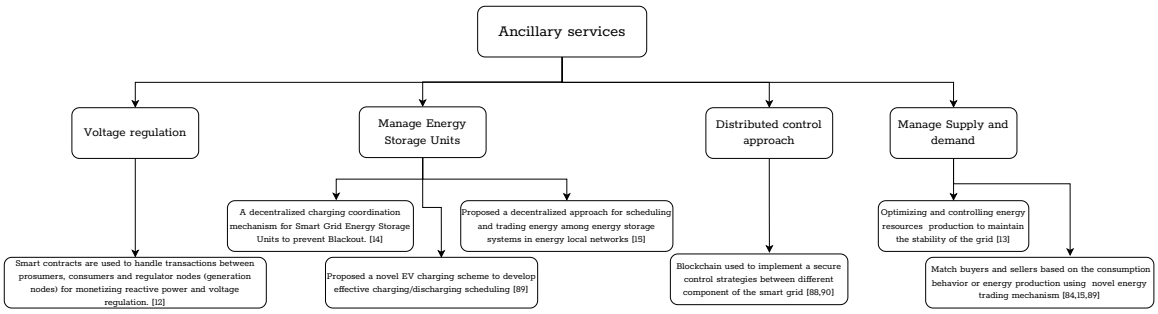

Figure 3.2: Semantic Diagram for Ancillary Services.

transmission infrastructure relative to local demand.

3. **Black Start Service**: Some generators need start-up power provided to them (e.g, a gas turbine needs outside power to bring the turbine up to adequate RPM before fuel can be injected). Generators who can restart their facility with no outside source of power are contracted to provide this power in the event of a total system blackout.

4. **Load Shed Services for imports**: LSSi systems permit AESO to shut down power flow to selected high-demand consumers, when necessary to balance energy demand with available supply (e.g, in the 2021 summer heatwaves, energy demand came very close to the maximum available supply. If it had exceeded that limit, AESO would have contractually been able to reduce total demand by cutting power to very large sites).

Blockchain and smart contracts provide transparency, traceability, and security in the system. These features can be used to design a robust and secure decentralized control system that enhances the reliability of the grid or optimizes and controls energy resources [74]. The blockchain provides a platform to track energy produced and consumed at each endpoint. Therefore, a decentralized control system can use this information to manage and prioritize supply and demand, leading to the power system's stability [83]. Smart Contracts are potentially designed in this kind of control system to handle secure transactions between consumers and producers, monitor

users' behaviour and data, define rules, and handle the decision-making process. Figure 3.2 shows a semantic diagram that classifies the smart contract design and usage in the ancillary service operations.

As the amount of PV generation in the grid increases, local microgrids can experience voltage instability at different hours of the day. During afternoons, the production is relatively high, and the household consumption is low. Danzi et al. [13] proposed a novel control schema using blockchain and smart contracts, which considers a subset of Distributed Energy Resources (DERs) to act as voltage regulators and reduce their power outputs over control periods. To balance the participation of DERs in voltage regulation, they consider incentives for DERs joining the regulating subset, which will be paid by the DERs that are not in the regulating subset and operate at their full capacity. The decentralized trading mechanism is also capable of managing sources and demand and helping suppliers to sell any excess power produced [87].

Energy storage units (ESUs), including stationary and mobile batteries, represent a powerful emergency backup during electricity outage events. ESUs can be used to overcome the disruption of renewable energy sources, allowing for a high integration level of eco-friendly energy sources [114]. In addition, the stored energy can reduce the stress on the power grid and enable effective demand response during peak load events. Furthermore, ESU owners can purchase energy from the grid during low tariff periods and use it during high tariff periods, which reduces the customers' electricity bills. However, energy trading among the storage systems becomes more complex when the number of participants increases [15]. Yang et al. [15] introduce an Automated Demand Response (ADR) framework for scheduling and trading energy among energy storage systems in energy local networks instead of trading electricity over long distances over complex meshes. They propose a price-incentive game-theoretic model to coordinate responsive executors' consumption behaviours for Energy Storage Systems (ESS). In order to match buyers and sellers, they use a schedulable ability

evaluation system using smart contracts.

Uncoordinated charging of ESUs may lead to an imbalance between charging demand and energy supply, resulting in instability of the overall grid [115]. Therefore, Baza et al. [14] proposed a decentralized charging coordination mechanism for Smart Grid ESUs to prevent a blackout. A smart contract receives each ESU's charging request, including its power demand, time-to-complete-charging (TCC), and the battery state-of-charge (SoC). It uses the Knapsack algorithm to select the ESU with the highest priority to be charged in the present time slot and defers the ESUs with lower priorities. The smart contract implements the rules to ensure that the charging schedules are computed correctly. Liu et al. [89] proposed a novel EV charging scheme based on a decentralized blockchain-enabled Smart Grid system to develop effective charging/discharging scheduling algorithms for efficient grid operations that minimize power fluctuations in the grid and the overall charging cost for EV users. For this purpose, they present an Adaptive Blockchain-based Electric Vehicle Participation (AdBEV) that uses the Iceberg order execution algorithm to execute the information posting and decision-making process. Mhaisen et al. [90] proposed a novel distributed control approach to enable collaborative and secure operations among Battery Energy Storage Systems (BESSs), improving Smart Grids' efficiency and reliability. They implemented control strategies as smart contracts and deployed them on a distributed network of BESS nodes. The experimental results show that smart contract enabled control is more robust than traditional schema to cyber attacks.

Silvestre et al. [12] present a comprehensive framework for microgrids' technical and economic management using blockchain considering ancillary operations, particularly the voltage regulation service. A blockchain is used to provide shared voltage regulations services to a microgrid and calculates the loss and reactive support. Smart contracts are used for automatic interaction between production and consumption nodes, and writing active and reactive power transactions in the blockchain.

Thomas et al. [88] used smart contracts to make agreements for shared control of

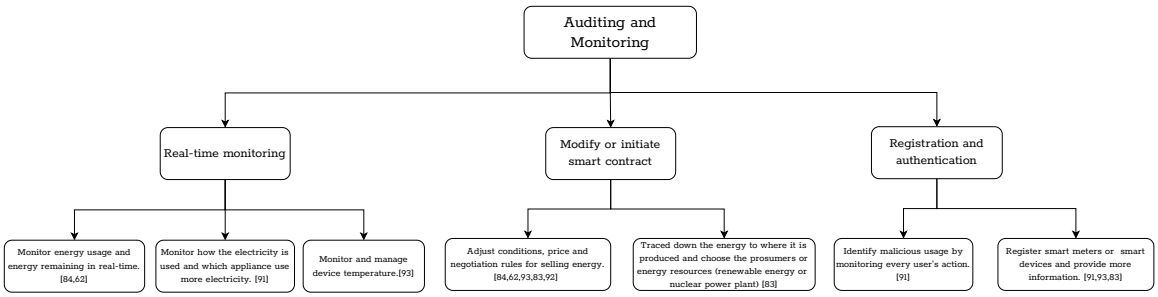

Figure 3.3: Semantic Diagram for Auditing and Monitoring.

energy transfer processes for a medium-voltage direct-current (MVDC) link, which is potentially responsible for system frequency and stability. In particular, smart contracts evaluate the cost and computational requirement for the agreement of control instructions.

3.2.3 Auditing and Monitoring

Studies in the area of decentralized Smart Grids also focus on monitoring and auditing via smart contracts, as shown in Figure 3.3. Monitoring in this context means smart contracts that monitor devices, users, and data to detect malicious behaviour or prevent threats to the system. Auditing refers to smart contracts that help customers track their energy usage in real-time, initiate contracts, or audit their preferences such as price, time, etc.

Gao et al. [91] present a system that protects consumers' data recorded and transferred on the Smart Grid using a tamper-resistant ledger while the users can transparently monitor the system. The smart contracts are designed to identify malicious usage of electrical power and electrical data. Whenever they find any malicious usage behaviour by a user, they revoke electrical power access and alert the user. Kang et al. [84] proposed a renewable energy trading platform using blockchain where each household is equipped with smart IoT devices and energy storage systems (ESS). Consumers can monitor energy usage and energy remaining in real-time. If a smart node detects insufficient energy, the consumer submits a transaction with the initial

price and the transaction process to register a purchase. Then prosumers can register a smart contract, including prosumer conditions for selling energy and price. Suther et al. [62] proposed an energy trading platform based on blockchain and smart contracts. They used blockchain to record transactions and trade history securely. The proposed trading platform provides an interactive user interface for trading energy locally among the members. In addition, a deviation settlement mechanism is applied to handle and modify negotiated trade. Yang et al. [93] proposed a mechanism called DMSM (Data Monitoring and Sharing Mechanism) based on the consortium blockchain. In this scheme, the smart devices are connected to smart contracts and send the report and notification to the users about the information such as device power or temperature. Then, users can deploy or adjust the smart contract through their smartphones. Finally, the electrical data in the Smart Grid is encrypted and stored in the cloud. The encrypted data can be shared with data-sharing nodes, like vendors or other entities, that require the data through key sharing. To increase privacy, they use BLS short signature and Paillier encryption algorithm to ensure the confidentiality of node interaction. Li et al. [92] propose a distributed renewable energy system for exchanging energy information in real-time between heterogeneous end-users. They present a non-cooperative game model to model inter-sectoral interactions between heterogeneous users, including residential, commercial, and industrial users. Smart meters periodically collect power consumption and production for each prosumer and stores them on a local grid-aggregator. Micro load management devices collect data from local grid-aggregators, calculate the dynamic electricity price and record it on the blockchain. Smart contracts are responsible for maintaining the stability of the power system. The designed smart contract will be initiated with the usage preference of each user. The dynamic price recorded by the M-LMDs triggers the execution of the smart contract. The detail of smart contracts implementation is not presented in this study. Yuhong Li et al. [83] presented a blockchain-based architecture for the Smart Grid, in which the consumers can be directly involved in

the system and trace the details of energy they have consumed from the grid. They can initiate a smart contract that contains each user's usage preference.

3.2.4 Cybersecurity

Within the context of blockchain applications in the Smart Grid, the cybersecurity literature tends to focus on mitigating one of two threats: unauthorized use of personal data, or the security of energy transactions (either utility-to-consumer or peer-to-peer). The former is a major concern because numerous studies have shown that patterns of energy usage can often be mapped to the identities and activities of consumers in a private home at a given moment in time. This needs not be restricted to cases where appliances are individually monitored; the field of non-intrusive appliance load monitoring has matured to the point that fluctuations just on the main power feed into a home are sufficient to reveal much of the occupants' activities [116]. Protecting the security of energy transactions, meanwhile, addresses necessary characteristics such as authorization, integrity, non-repudiation, and auditable fulfillment of an energy contract. The fundamental difference between these two strands of research appears to be the presence or absence of a middleware layer that protects data privacy.

Studies that focus on privacy protection generally rely on ensuring that a data requestor (e.g. a power utility studying demand patterns in its network) can only access data that the owner (i.e., the consumer) explicitly permits them to access; and that, furthermore, the requestor only receives anonymized, summarized reports concerning that data. Access to the raw data itself is not provided, so as to ensure that the owner's privacy cannot be breached by accident, malicious intent, or external penetration of the requestor's IT systems. Figure 3.4 is a good example of this approach [75].

In Figure 3.4, a utility wishes to access consumer data. This might be as simple as remotely reading a smart meter for billing purposes, or as complex as an in-depth

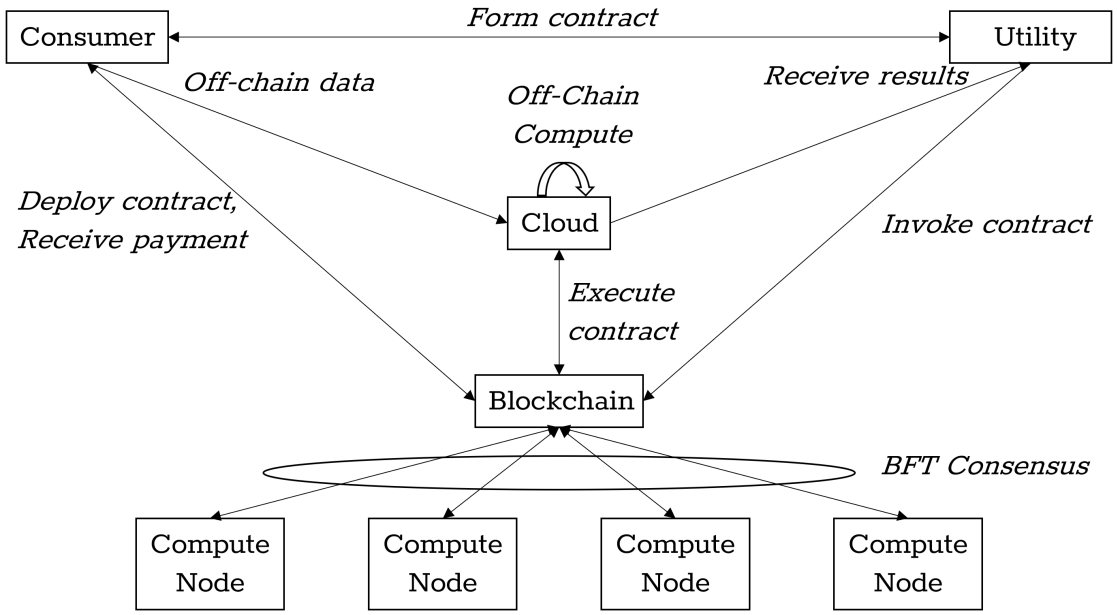

Figure 3.4: Privacy-Preserving Data Access via Smart Contracts [75]

study of demand patterns (e.g., system planning for widespread electric vehicle adoption). The consumer will securely store their data in an off-chain database (assumed to be cloud-based). The utility and consumer will negotiate a smart contract (implemented in the appropriate mechanism for the underlying blockchain) defining what data will be provided to the utility in what form, and what compensation will be provided to the consumer for doing so. The consumer deploys this contract to a blockchain, and it is validated in a consensus protocol. The utility can invoke the contract, which triggers a cloud-based off-chain computation (defined by the contract and NOT under the utility's control) involving the consumer's data. Only the result of this computation is returned to the utility, as part of an atomic transaction that also sees the consumer paid their compensation [75]. Another approach to securing raw data is to deliberately add noise to the data, as in Gai et al. [94].

Dorri et al. [76] presented a framework that allows energy consumers and producers to directly negotiate an energy price using blockchains to preserve privacy, by using a routing method based on the destination public key. A two-phase commit protocol

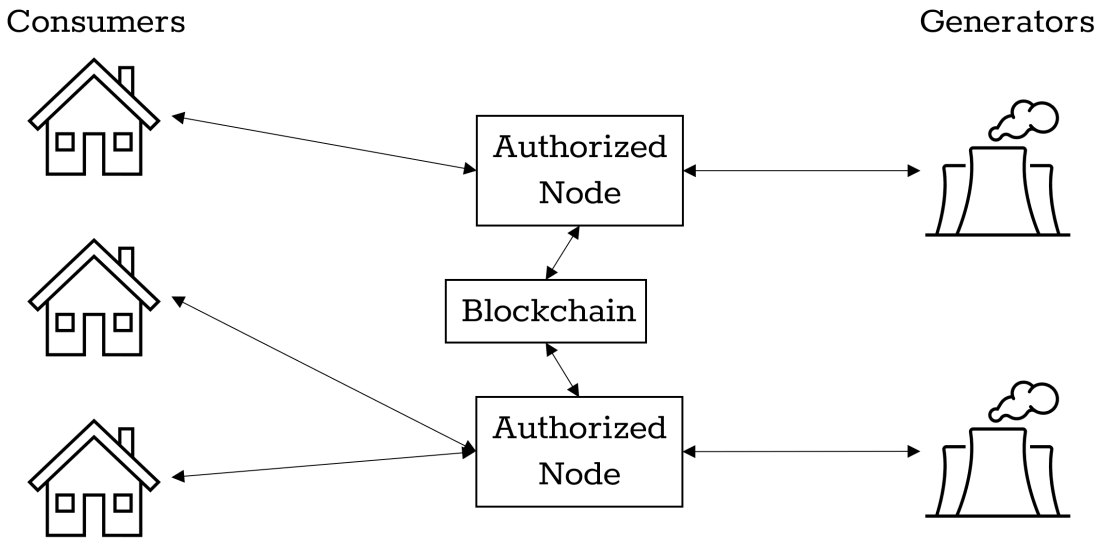

Figure 3.5: Blockchain-Based Energy Trading [100]

ensures contract fulfillment and settlment. They evaluate a Proof-of-Concept (PoC) implementation in a second paper [117]. They also investigated a framework with removable blockchain networks to improve users' scalability, throughput, latency, and privacy [97]. J. Wang et al. [98] proposed an anonymous authentication and key agreement protocol for a blockchain and smart contracts.

The second research theme, securing energy transactions, is very much a classic application of blockchains. In the specific context of energy systems, the basic approach of [100] illustrates several key concepts. In Figure 3.5, consumers and generators negotiate smart contracts for energy delivery. Each generator creates at least one Authorized Node, which can form contracts and interact with the underlying blockchain (including executing the consensus algorithm). In this specific example, an Ethereum blockchain is used, and consensus is achieved using Ethereum's Proof of Stake algorithm. That algorithm relies in part on the Ether coin, and so [100] proposed that contract settlement also be carried out in Ether coins. While Figure 3.5 depicts the coins on the links between physical entities, the actual coins are of course simply recorded on the blockchain.

Gai et al. [95] proposed a permissioned blockchain to counter fraudulent energy usage, communication interference, and data center attacks. Lombardi et al. [96] used smart contracts to implement energy trading and auction transactions. They also detect and isolate smart meters that are vulnerable to tampering. Gao et al. [91] used smart contracts to detect malicious usage of electrical power. W. Wang et al. [99] suggested a framework based on the combination of blockchain, Elliptic Curve Cryptography (ECC), dynamic Join-and-Exit mechanism and batch verification to solve some identity authentication issues in the Smart Grid. They considered smart contracts for interaction requirements such as a query for participants, user registration, signature verification, and user logout. Aung et al. [101] focuses on the privacy of homeowners' data, but they also developed an access control mechanism for an emergency state wherein Home Service Provider (HSP) staff are granted additional access during a service vsit.

3.3 Data Storage

In many studies discussed in Section 3.2, smart contracts gather and record data from smart meters on-chain for further processing. Smart meters, however, report readings at a high rate (often once per minute per meter), and in a realistic energy market, there would be a large number of them representing consumers and prosumers. This seems to be a poor use of blockchain network resources because large files cannot be efficiently stored on blockchains. Wüst and Gervais [118] discussed whether a blockchain is suitable for a given purpose. They provide a structured methodology to determine whether a blockchain, as a unique type of distributed database, is an appropriate technical solution to solve a problem with a hint of the proper network type (permissioned or public). As an immutable and tamper-proof ledger, blockchains are a valuable storage solution; however, they have their disadvantages. Everything stored on the ledger is replicated on all peers, and it will remain permanently, which requires a lot of storage space. Besides, each node on the blockchain may not need to

view every file that is stored on the blockchain. Hence, storing data on-chain could thus be a costly and inopportune solution, and it is critical to determine what data is worth keeping on-chain and what should be stored off-chain. In particular, blockchain is not an appropriate solution for streaming data as there is no means to discard data that ages out of relevance. In the following, we discuss three potential solutions to solve the problem of data storage.

3.3.1 Removable Ledger

Dorri et al. [97] proposed an approach in which the required data for managing energy trading in real-time operations are stored on a temporary blockchain, where the transactions are stored on the ledger for a specific period of time. This reduces the size of the ledger, as well as helps with scalability, throughput, latency and privacy of users. This study suggests two blockchain network layers: temporary chain and main chain. Transactions stored in the temporary chain will be removed once they expire; however, the main chain maintains the hash of all transactions ensuring the traceability of the operations.

3.3.2 InterPlanetary File System

InterPlanetary File System (IPFS)[1] is a peer-to-peer distributed data storage, and file-sharing system that seeks to connect all computing devices with the same system of files [119]. Files are sliced into multiple parts and stored across multiple nodes, and they can be tracked through their hash values that can easily be stored on a blockchain. Because of the low latency to access the data and high throughput, it appears to be a reasonable alternative off-chain distributed file system [120]. To increase the security of sensitive data on the IPFS, Steichen et al. [121] proposed a modified version of IPFS that provides access-controlled file sharing. In this approach, the smart contract maintains the access control list and interacts with IPFS whenever

[1] https://ipfs.io/

a file is uploaded, downloaded or transferred.

In energy studies, Kumari et al. [48] proposed storing smart meter data, including energy generation and consumption and the details of energy records on IPFS. The stored data can be accessed through the participants' hash key. Then, only the hash key is stored on the ledger. Aung et al. [101] also employ digital signature coupling with IPFS to handle emergency calls from a smart home to home service providers while protecting the homeowner's privacy.

3.3.3 Store Synopses and Essential Facts

An off-chain database can be used to store streaming data for a limited period, while synopses of the data are generated. These synopses are then stored on-chain, while expired off-chain data can be discarded. Thus, we have a permanent and tamper-proof series of synopses of the data stream on the blockchain. Other important data that may be stored on the blockchain are related to legal contracts, including the result of bids and winners, the price of energy, the payments, and fulfillment information.

3.4 Interoperable Blockchains

The Smart Grid layered architecture includes multiple communicating networks passing large amounts of heterogeneous data between them at various timescales [122]. Using a single blockchain for such a complex system may not be an effective approach [123]. Alternatively, it is possible to design systems composed of multiple interoperable blockhains [124] that automate energy management transactions and secure a complete operations log. A different blockchain platform with appropriate properties can be employed for each layer [52]. Another reason that interoperability has been investigated is to improve scalability by running multiple chains in parallel and offloading transactions into multiple blockchain networks. This section, therefore, discusses studies mainly focused on developing multiple interoperable blockchains [124].

3.4.1 Extending the Application Scale

Li et al. [123] propose a framework utilizing four permissioned blockchains, each tailored for a distinct purpose. They are responsible for market initialization, energy trading, state estimation and market settlement, respectively. The market initialization blockchain includes smart contracts automating network configuration and market rules for energy transactions. The energy trading blockchain handles generation offers and consumption bids submitted by market participants. The state estimation blockchain shares data, including operating states at participants' physical boundaries, and it utilizes smart contracts to estimate the system operating state for energy transfers. The market settlement blockchain calculates and stores the results of financial settlement and the changes in participants' reputation scores. The framework offers decentralized coordination between untrusted energy market participants, recording and sharing energy and financial flows between transacting microgrids, and payment transactions. For example, the market settlement blockchain communicates with the market initialization blockchain to receive the pseudonym of each transacting microgrid and the power network configuration to settle scheduled and actual energy transfers.

Liu et al. [125] introduces an Ethereum-based dual-chain architecture. The first chain is the local energy trading blockchain (LETB) that records and publishs information on the local electricity market. The second chain is the regional renewable energy trading blockchain (RETB) that records information about the renewable energy producers. Their approach has three phases: initialization, local power trading and renewable energy trading. In the first phase, each entity joins the corresponding LETB and RETB blockchain. Then in the local power trading phase, the smart contract at LETB blockchain arranges power distribution plans based on the demand of consumers and producers. After that, the reputation value of each producer based on consumers' feedback is modified by the smart contract, and producers get incentives.

Furthermore, if local power distribution fails to meet consumers' demand, the renewable energy trading phase is initiated. The smart contract matches the renewable energy producers to provide the remaining power demand. If this still fails to meet the load power plan, the local grid will supply the remaining power as a backup.

3.4.2 Scalability and Performance

Ochôa et al. [126] propose a blockchain architecture that uses sidechains to make a scalable and adaptable system for Smart Grid applications and ensure consumer privacy. Sidechains use a Cross-Chain Communication Protocol (CCCP). It is a mechanism for two running blockchains to interoperate in which the main blockchain (mainchain) holds the second blockchain as an extension of itself. The mainchain keeps a ledger of assets, and it uses CCCP to connect to the sidechain. The proposed framework uses three different blockchains and integrated them with the Open Smart Grid Protocol (OSGP) and smart contracts. Each blockchain in the system is responsible for a different application in the Smart Grid: tracking users' privacy preferences, storing users' data, and contract fulfillment and settlement.

Kong et al. [127] introduce a framework for power systems that runs simultaneously on multi-chain networks to enhance the scalability and throughput of the blockchain network. The multi-chain approach is proposed to overcome the scalability problem in the BFT protocol due to the high arrival rate of new messages. This solution divides the blockchain authority system into multiple sub-networks called Blockchain Autonomy System (BAS), where each maintains an independent ledger. BAS is responsible for data collection, broadcasting and sharing. Block mining and ledger maintenance in separate BASs are independent and parallel. The sensor nodes broadcast measured data to the base station nodes, and in order to identify different BASs and base station nodes therein, a 32-bit BAS ID is assigned. Hence, all base station nodes belong to the same BAS hold the same BAS ID.

3.5 Conclusions

As blockchain has matured, it has received more and more attention as a resilient platform for Smart Grid operations. Some research and surveys have investigated and developed specific areas of distributed energy management applications through blockchain, but the design and performance of smart contracts in general have never been analyzed or investigated. In this work, we present a systematic review focusing on the smart contract design and capability of blockchain-based systems in Smart Grid. Based on the Smart Grid application domain, we categorized our primary sources into four fields: market operations, ancillary services, auditing and monitoring, and cybersecurity. We explored the different designs of smart contracts in each field and investigated the performance evaluation of blockchain-based systems in energy systems. We find that data storage and blockchain interoperability are major concerns in these areas, and we discussed available solutions for them.

Chapter 4

A Scalable Blockchain-based Smart Contract Model for Decentralized Voltage Stability Using the Sharding Technique

This chapter discusses the theory and design of our proposed blockchain-based voltage stability monitoring and control model. We first provide essential background in voltage stability and Volt-Var control techniques, and then we describe the Decentralized Voltage Stability (DVS) algorithm. We then discuss the design of our smart contracts and our proposed model architectures on the blockchain. Finally, we introduce and evaluate the sharding technique in the context of our smart contracts.

4.1 Introduction

Voltage stability has always been a vital concern in the power grid, and it grows ever more important with the growing presence of renewable energy sources and EV chargers. The advent of new Intelligent devices, such as Intelligent Electronic Devices (lEDs), Phasor Measurement Units (PMUs) and smart meters help the Smart Grid system to enable various fine-grained monitoring and control techniques for keeping the voltage stable in the grid. However, these devices also create vast amounts of data that threatens to overwhelm centralized control systems. Therefore, many

studies have investigated decentralized monitoring and control approaches to overcome this problem [17] [128] [129] [130]. In these studies, the energy system is split into multiple local grids. The control algorithms will principally operate on localized voltage measurements but must still ensure system-level stability.

Blockchain, as one type of decentralized platform, can provide security, resilience, and traceability for the monitoring and control of system-level objectives such as voltage stability protection. However, the performance and scalability of blockchain for real-time applications, such as voltage stability monitoring and control system, have not been empirically tested. We present a practical, scalable, decentralized voltage stability algorithm based on blockchain and smart contract technology to evaluate and investigate blockchain technology performance and capability. To the best of our knowledge, this study is the first of its kind.

Our contribution is the design and evaluation of a blockchain-based smart contract implementation of an existing distributed voltage stability control algorithm (DVS [17]), which is in contrast to the original DVS design. We used the HyperLedger Fabric [18] platform to implement the monitoring and control algorithm, and manage resources in the electric power system by smart contracts. For power simulation, we used the Matpower package [1] [19] in Matlab. We connected these two simulation platforms via a Restful API to transfer the grid's data to the blockchain and evaluate our system. We tested our implementation using the benchmark IEEE 30 bus topology [131].

In our initial design, we showed that smart contracts can be used to implement and utilize complex algorithms for a real-time system. Then, we applied the sharding mechanism to improve the system's performance and improve our model's scalability for larger networks. In this model, by growing the Smart Grid network, one or multiple local controllers can be assigned to different shards within the network, and a subcommittee is selected for each shard, which allows many more transactions to be

[1] https://matpower.org/

processed in parallel. For evaluation, we used the Hyperledger Caliper benchmarking tools [20], and tested our system with a different number of shards, which shows that our solution scales linearly with the number of shards compared to an unsharded approach.

The rest of this chapter is organized as follows. The first three sections provide necessary background in voltage stability, Volt-Var control techniques, and the DVS algorithm. Section 4.4 presents the design details and architecture of our proposed smart contract model, and then we investigate the sharding mechanism for improving the scalability of our system. Section 4.5 explains the implementation and deployment details of our two models, presents the evaluation steps, and compares the result of our base model versus the sharding model. Section 4.6 summarizes the conclusions of this work.

4.2 Voltage Stability

In the electric power grid, power demand (load) must be less than or equal to the total available power. If the load in a given region exceeds the available power and no corrective actions are taken, the area voltages will become unstable and even collapse. This can severely damage electric systems, and so a common response has been to cut off the power flow to that region completely, resulting in a blackout. Voltage collapse or instability is a dynamic circumstance involving many nonlinear power system components, on a time scale that can range from seconds to hours [132].

Voltage stability refers to the ability of each node to maintain or restore an acceptable voltage limit after suffering disturbance. The voltage stability limit is the maximum voltage value that the system can handle as the total value of the load power; when all the load values of the system reach a certain level and if an additional load is added, the load node voltage might fall dramatically and lead to voltage collapse.

Voltage instability occurs when the system can not meet the voltage stability limit

in the system. The node voltage of the system will either increase or reduce in several seconds or several minutes after the normal operating power system. The main reason for voltage instability in the system is the lack of reactive power, on-load tap changer dynamics and increased loading on transmission lines. Voltage collapse is caused by significant disturbances which lead to line outage, generator unit tripping of electric elements, and potentially a cascading blackout that results in an irreversible declining process of local or global power grid voltage and the loss of power of many loads. In other words, voltage collapse refers to the load voltage of the system that is below the acceptable limit value due to voltage instability [133]. Voltage instability and voltage collapse are two terms that can be used interchangeably.

The voltage stability can be classified based on the type of disturbance and time scale. Considering the time scale, voltage stability can be divided into long-term and short-term stability due to their period and time length. The timeframe of interest in short-term voltage stability is the order of a few seconds after the initial disturbance, mainly studying the high-voltage direct current (HVDC) converter and induction motor. In contrast, the long-term voltage instability lasts longer, several minutes to dozens of minutes, a situation that usually involves the generator excitation current limiter and transformer tap adjustment, and so on [133] [134]. Moreover, voltage stability may be classified as either a small disturbance or a large disturbance. Small disturbance voltage stability refers to the ability of the power system to maintain acceptable voltages of buses after minor disturbance, such as load rising by a small amount. Disturbances such as losing a generator, short circuit, line outage, or system failure are considered as large disturbances.

To confront voltage instability, generally, two control strategies are used: preventive strategy, which ensures that the pre-disturbance system maintains sufficient margin to the stability boundaries; and corrective strategy, which is applied after disturbance to restore the system state to a stable point. The preventive strategies will be effective if proper corrective actions are executed before the voltage collapse. Therefore, early

prediction of voltage instability and timely execution of appropriate corrective actions are critical to prevent voltage instability.

From another point of view, these methods may use local or global measurements. Methods that only use the voltage and current measured at the local bus are called local measurement approaches, such as Thevenin's impedance matching condition, and its derivative [135]. These methods are fast, but their main drawback is the need to determine the Thevenin equivalent parameters accurately. Unlike local methods, global methods use the data gathered from all over the system. Global methods have more ability to analyze and identify weak buses and the proximity of the system state to the stability boundaries since voltage instability can involve many components interacting nonlinearly. The main drawbacks of these methods are that they are themselves highly nonlinear and computationally expensive. These days, the advent of wide-area measurement systems (WAMSs) has opened up new avenues for global voltage stability protection. In the WAMS methods the entire power system becomes observable by precisely and synchronously measuring currents and voltages at different system points at a high rate of data transfer. In other words, WAMS makes the dynamic behaviour of the power system visible for applying online control to the system. WAMS uses a measurement system composed of strategically placed phasor measurement units (PMUs) that can monitor a critical area's current status in real-time. Various studies investigate strategic locations for placing PMUs decided in such a way that the number of locations is minimized while the critical area remains completely observable [136] [137][138].

A PMU is a device used to estimate the synchrophasor, frequency, and the rate of change of frequency (ROCOF or df/dt) of the acquired voltage or current waveforms. A synchrophasor is the magnitude and angle of an electrical phasor quantity, such as a cosine signal of voltage/current measured at an absolute point in time [139]. Synchrophasor traffic has different latency requirements, ranging from 20 ms to 200 ms based on the applications. Depending on the number of phasor measurement units,

the word length, the number of samples, and the frequency, the required bandwidth is about a few hundred kbps. The system's frequency is internally computed at a higher sampling rate and reported at 30–60 samples per second, which has great potential for building dynamic monitoring systems. A fixed sampling rate is used with a synchronized global positioning (GPS) system, and all the measurements are coupled to a timestamp allowing for the alignment and synchronization of PMUs spanning an entire interconnection. Phasor Data Concentrators (PDCs) collect and transmit the data of several PMUs and aligns the measurements in time. It can also be configured to store and archive data or perform calculations on the measurements.

Increasing demand, the adoption of renewable energy sources, and contingency impacts on interconnected power grids increase the need for real-time control in the power system. PMUs provide a suitable platform for real-time measurements and design strategies that allow real-time controls, such as adjusting voltage regulators or controllable series/shunt compensators, to arrest voltage instability at its onset. Studies during the past years proposed different techniques to solve voltage stability problems using PMU data for online prediction of dynamic voltage instability such as real-time voltage stability index [140], Machine Learning-based model [141] [142] [143], Metaheuristic algorithm [144], etc.

4.3 Volt-Var Control

Volt-Var control for Decentralized energy Resources (DER) refers to power converters used to enhance the voltage's stability and reliability in the distribution system. Voltage regulating devices are usually installed at the substation to adjust the voltage, depending on the loading condition of the feeders. Reactive compensation devices (i.e., capacitor banks) are used to adjust the reactive power flows (meaured in units of Volt-Amps Reactive, or VAR) throughout the distribution network.

A static VAR compensator (SVC) is a multimachine system for providing fast-acting reactive power on high-voltage electricity transmission networks [145] and

many studies and control approaches used this device to compensate for voltage instability in the system. The SVC can adjust reactive power over an unlimited range without any time delay in minor steps. SVCs are used in two main situations: Transmission SVC, which connects to the power system to regulate the transmission voltage, and Industrial SVC, which connects near large industrial loads to improve power quality. The main advantage of SVCs is their near-instantaneous reaction to changes in the system voltage. They provide the required reactive power due to simple mechanically switched compensation schemes. SVCs are generally more reliable than similar devices such as static synchronous compensators (STATCOM) and Unified Power Flow Controllers (UPFC). However, static VAR compensators are more expensive than mechanically switched capacitors. Therefore, a combination of them is used by many operators. In this case, the static VAR compensator supports fast changes, and the mechanically switched capacitors are used to provide steady-state VARs. In this literature, we refer to these combined devices as Volt-Var Compensators (VVCs).

4.4 The Decentralized Voltage Stability (DVS) Algorithm

Nowadays, ensuring voltage stability is a challenge due to the increasing number of renewable power sources, increasing loads, environmental limitations on power system expansion, and high competition in the energy market. Voltage stability maintenance is an active process, which can be organized in a centralized or decentralized fashion. In the centralized voltage stability algorithms, a control center monitors the state of the power grid (or a subregion), commonly represented as one or more voltage stability indexes. These indexes essentially capture the grid's current safety margin against a voltage collapse. When stability is threatened (the indices approach dangerous values), the control center intervenes. Stored power from capacitors (purposefully charged for such needs) could be released, large-scale power customers might see

large machines idled, or additional generation may be brought on-stream. However, the volume and velocity of data received at the control center means that it might not react swiftly to instabilities in a smaller region. Decentralized algorithms, on the other hand, should be able to react much more swiftly to local instabilities. In this case, a voltage stability index would be calculated just for a limited area of responsibility, implying a far lower volume of data to monitor and process. The challenge, of course, is that a local algorithm will have far fewer resources available to address any instability. A mechanism for escalating interventions so as to call on resources beyond the local area will be needed [16].

Several studies have proposed or investigated different decentralized monitoring and control techniques, but none of them used blockchain and smart contracts [11] as a distributed computing platform. Blockchain in this kind of system can increase trust, security, transparency, and the traceability of data shared across a power system network. We furthermore hypothesize that it provides a robust computational framework to implement local and distributed voltage stability algorithms. To test this hypothesis, we will implement one such algorithm as a set of smart contracts. From the existing literature, we selected the DVS algorithm [17]. First proposed in 2016, DVS includes both monitoring and control algorithms in their architecture. The authors validated the algorithm in simulations using the IEEE 30, 57, 118, and 300 bus topologies [16] (these are standardized power network simulations, with a varying number of connnections, used as research testbeds in power delivery systems research. Physically implementing and testing a new algorithm on the actual power grid is obviously inadvisable, and physical laboratory testbeds are either too simple to replicate actual power-system dynamics, or else prohibitively expensive.) The DVS algorithm is as follows:

Initial Grouping and Group Formation:

The DVS algorithm starts with an initial grouping method, which splits the power system grid and resources into multiple small groups. Each group is a set of nodes or

substations aggregated based on their electric distance and network sensitivity. The substations or nodes in our blockchain-based model are able to instantiate smart contracts. They record and emit synchrophasor measurements from phasor measurement units (PMUs) to compute voltage stability indices [140] [146]. They can take control actions based on those measurements using local reactive power sources or Volt Var Compensators (VVCs). At a minimum, each group is assumed to have a few reactive power sources or VVCs available, as well as at least one transmission line and load and generator.

The buses in the local group can be classified into one of three categories: load bus, tie or boundary bus, and generation bus. Load buses have a load connected to them. Generation buses have power sources or generators, and the tie buses are the interconnections among groups. For simplicity, they split each tie line in half and replace it with either a virtual load bus (virtual PQ bus) or a virtual generator (virtual PV bus) based on the power flow direction to represent power grid connections outside of the local group.

The power flow directions are computed by PMU measurements. Equation (4.1) shows the calculation of power flows out of the tie bus. If the real part of S has a negative value, then the line is replaced as a virtual generator; if it has a positive value, the line is replaced as a virtual load.

$$\begin{aligned} S_{ij} &= V_i * I_i \quad Power\ leaving\ from\ Bus\ i. \\ S_{ji} &= V_j * I_j \quad power\ leaving\ from\ Bus\ j. \end{aligned} \tag{4.1}$$

DVS Monitoring Algorithm:

The DVS monitoring algorithm estimates a voltage stability index (VSI) using the proposed Thevenin's Equivalent approach in [140]. Based on Thevenin's theorem, we can simplify any complex circuit to an equivalent circuit with just a single voltage source and series resistance connected to a load. In the above mentioned paper, the equivalent voltage and impedance are calculated based on each group's information

for representing the external system connected to each load bus. First, they create an admittance matrix for each group for representing the network topology. Equation (4.2) shows an admittance matrix, in which the G denotes generation bus, L represents Load bus, and T shows the tie bus. Therefore, Y_{GL}, Y_{GT}, Y_{TL}, Y_{TT} , Y_{TG}, and Y_{LL} are the admittances between generator to load, generator to tie line, tie line to load, tie line to tie line, tie line to generator, and load to load, respectively.

$$Y = \begin{bmatrix} Y_{GL} & Y_{GT} & Y_{GG} \\ Y_{TL} & Y_{TT} & Y_{TG} \\ Y_{LL} & Y_{LT} & Y_{LG} \end{bmatrix} \tag{4.2}$$

$$Z_{th} = Z_{LL} = (Y_{LL} - Y_{LT}Y_{TT}^{-1}Y_{TL})^{-1}. \tag{4.3}$$

$$\begin{aligned} V_{thj} &= ((\frac{S_{Lj}}{V_{Lj}})^* * Z_{thj}) - V_{Lj} \\ j &= 1, ..., n \; n = number\ of\ load\ bus. \\ S_{Lj} &= complex\ power\ flow\ out\ of\ bus\ j. \\ V_{Lj} &= voltage\ magnitude\ of\ load\ bus\ j. \end{aligned} \tag{4.4}$$

Thevenin's parameters are calculated using equations (4.3) and (4.4). The V_{th} (voltage equivalent) and Z_{th} (equivalent impedance) are then used to approximate maximum active power (P_{max}), maximum reactive power (Q_{max}), and maximum complex power (S_{max}) for each load bus [140]. (Recall that alternating current power follows a sine-wave pattern, which is analyzed using complex values; reactive power is the imaginary component of the complex power vector.) The approximate P_{max}, Q_{max}, and S_{max} can be expressed by following equations:

$$
\begin{aligned}
P_{max} &= \sqrt{\frac{V_s^4}{4X^2} - Q\frac{V_s^2}{X}} \\
V_{th} = V_s &= source\ voltage\ magnitude. \\
Z_{th} = Z_L &= \sqrt{X^2 + R^2} \\
X = reactance, &\ R = resistance.
\end{aligned}
\tag{4.5}
$$

Similarly, $(P_{load} = P, Q_{load} = Q, S_{load} = S = P + jQ)$ are the real, reactive, and complex power values of load at each bus (please refer to [140] for a detailed derivation).

$$
Q_{max} = \frac{V_s^2}{4X} - \frac{P^2 X}{V_s^2} \tag{4.6}
$$

$$
\begin{aligned}
S_{max} &= \frac{(1 - \sin(\theta)V_s^2}{2\cos(\theta)^2 X} \\
\theta &= \sqrt{\frac{Q}{P}}
\end{aligned}
\tag{4.7}
$$

Using calculated maximum power for each bus, the VSI for each load bus is calculated as follows:

$$
VSI = Min(\frac{P_{max} - P_{load}}{P_{max}}, \frac{Q_{max} - Q_{load}}{Q_{max}}, \frac{S_{max} - S_{load}}{S_{max}}). \tag{4.8}
$$

The VSI for each bus is used in the DVS algorithm to detect buses with a weak stability margin; the DVS control algorithm is then used to prevent voltage collapse.

DVS Control Algorithm:

When a bus is found to have a weak stability margin, additional reactive power from the closest source (in terms of electrical distance) is injected into the network. In this algorithm, a Priority Index (PI) matrix is first formed for each group using the admittance matrix (magnitude value of admittance matrix). The PI matrix represents

an electrical distance between buses and helps to find the closest VVCs to the weak bus. Based on this matrix, the top priority is given to a VVC directly connected to the weak bus; and the next set of priorities is given to reactive power sources based on the ascending ranking of electrical distance. After selecting the closest reactive power, the required reactive power to adequately raise the stability margin at the weak bus is calculated using the Jacobian matrix. In Equation 4.9, V_{req} is the minimum acceptable voltage, $Q_{req}$is the required reactive power, and $V_{WeakBus}$ is the voltage magnitude of the weak bus.

$$Q_{req} = \frac{\delta Q}{\delta V} * (V_{req}) - V_{WeakBus}). \tag{4.9}$$

The $\frac{\delta Q}{\delta V}$ term represents the sensitivity of the bus where the VVC is located versus the weak bus. The sensitivity of each bus respect to other bus or its own is calculated by following equations using the Jacobian matrix:

$$\begin{gathered} for \;\; k, n = 2, 3, 4, ..., N \;\; N = \#\; bus.\; Y = admittance\; value \\ \delta = voltage\; angle \;,\; \theta_{kn} = admittance\; phase\; angle = \tanh(\frac{imag(Y)}{real(Y)}). \\ \frac{\delta Q_k}{\delta V_k} = -|V_k||Y_{kk}| \sin(\theta_{kk}) + \Sigma_{n=2}^{N} |Y_{kn}||V_n| \sin(\delta_k - \delta_n - \theta_{kn}) \end{gathered} \tag{4.10}$$

$$\frac{\delta Q_k}{\delta V_n} = -|V_k||Y_{kn}| \sin(\delta_k - \delta_n - \theta_{kn}). \tag{4.11}$$

For the tie buses, we need to calculate virtual voltage magnitude and angle and double the value admittance matrix as we split the bus in half. For this part, we first create an admittance matrix for each group, based on Equation 4.2, by considering the new admittance value for tie buses (Equation 4.12). Then we calculate the virtual voltage by dividing the sum of the voltage value of both sides of the tie bus by 2. Equation 4.13 shows the calculation of admittance, virtual voltage magnitude and angle.

$$\begin{aligned} &if\ we\ have\ load\ bus\ between\ load1\ and\ load2: \\ &Y_{12'} = 2 * Y_{12},\ 2^{'} = virtual\ bus. \end{aligned} \tag{4.12}$$

$$\begin{aligned} Voltage\ magnitude = V_{m12'} = \frac{(V_{m1} + V_{m2})}{2}. \\ Voltage\ angle = V_{a12'} = \frac{(V_{a1} + V_{a2})}{2}. \end{aligned} \tag{4.13}$$

As the admittance values show the connection between two points in a grid and there is no direct bus or connection between some of the loads and generators, the admittance value for these connections is zero. In the DVS algorithm, we chose the shortest path using the PI matrix and used the chain rule to calculate the sensitivity of buses that are not directly connected to the weak bus.

If needed, the control algorithm may run multiple times to compensate for the voltage stability at the target bus. If the DVS algorithm cannot find enough reactive power sources within the group to correct the stability margin, it then merges the group with an adjacent group, and calls on that group's resources as well. This process will continue, drawing in resources from a wider and wider area until enough additional power is injected to restore stability.

4.5 The Blockchain-based Smart Contract Design for DVS Algorithm

Blockchain technology can be leveraged when implementing the DVS algorithm. The blockchain model provides a secure platform for each group to communicate with each other and reach a consensus on which buses require control action, and how best to organize the same. Utilizing smart contracts enables us to automate these procedures, thus reacting faster to voltage instabilities; as noted previously, a voltage collapse can potentially happen in just seconds. At the same time, the security and immutability of the blockchain offer enhanced protection and auditability for critical

infrastructure. While other authors have explored blockchain systems in the past, they either do not use blockchain as a platform for decentralized voltage stability, or neglect the performance of their consensus mechanism if they do [17].

This study proposes and evaluates a blockchain-based smart contract model to implement the DVS algorithm. We furthermore address scalability; plainly, in a large-scale grid there will be many groups, and transactions between groups. The blockchain network underlying the grid could thus be overloaded. Therefore, we designed a sharding solution for our blockchain-based DVS algorithm. We formally discuss the design of our model for the sharding mechanism based on two consensus levels: shard-level consensus and mainchain consensus. This section will elaborate on our blockchain-based approach and workflow and then describe our sharding mechanism to improve scalability.

4.5.1 Workflow

In this subsection, we discuss the entire workflow of our proposed framework and elaborate on the network topology and smart contract details for DVS monitoring and control algorithms. First, we categorize the participants in our network, following the notation in Figure 4.1.

Clients:

We assume that multiple PMU devices are distributed inside each group for measuring phasor quantity. PMU devices estimate the magnitude and phase angle of an electrical phasor quantity, such as voltage or current, in the power grid. PMUs can report high temporal resolution measurements, up to 120 Hz [147]. Several papers investigated the optimal location of PMUs when the number of PMUs is limited [137][138], which is out of the scope of this paper.

In order to gather measurements, we assume one or multiple computing devices are responsible for aggregating the PMUs measurement in each group and instantiating relative smart contracts through the Fabric SDK. The Fabric SDK allows applica-

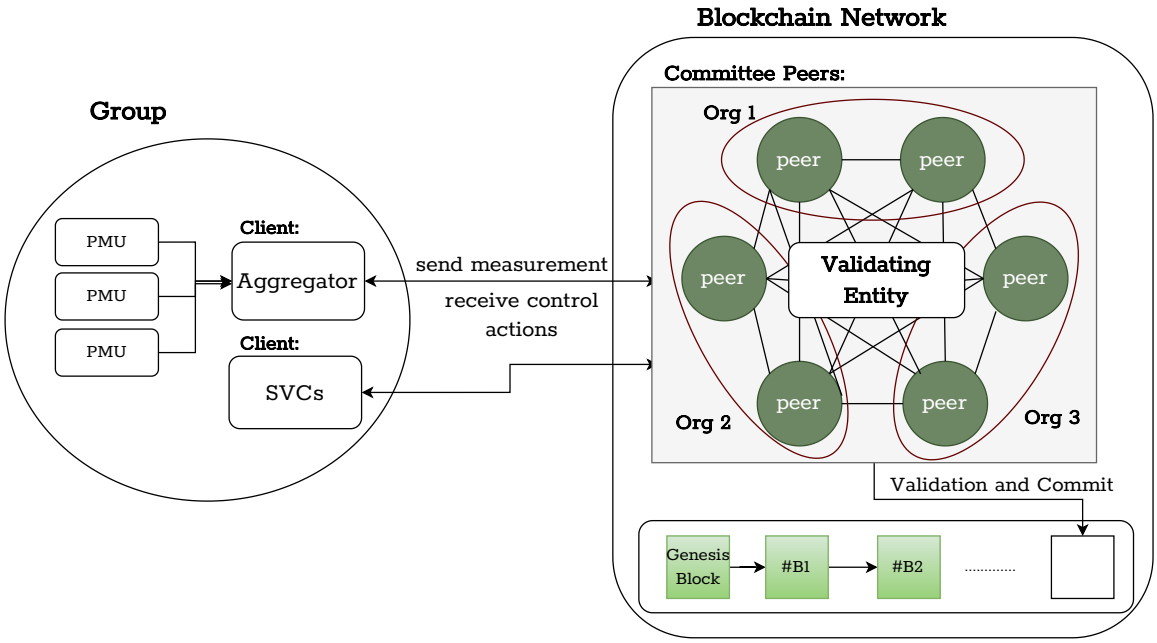

Figure 4.1: Blockchain-based architecture for the DVS algorithm - without the sharding mechanism.

tions to interact with a Hyperledger Fabric blockchain network via a simple API to submit transactions to the ledger or query the recorded data with minimal code. Furthermore, the reactive power sources and VVCs are also connected to the network and update their available resources on the ledger for control actions. In the case of need, smart contracts can send a control action to activate or deactivate VVCs on the grid. For simulating the client-side, we use the Matpower library in Matlab to run optimal power flow computations, and inject reactive power to each bus. For interacting with Fabric SDK, the related data of each group is converted to JSON. Then the data is sent via an HTTPS request to a RESTful Web-service to submit or query a transaction on the ledger.

Organization and Peers:

The HyperLedger Fabric blockchain network is built up from the peers owned and contributed by the different organizations. Each organization can have one or multiple peers responsible for various tasks and offering API services for clients. The smart contracts are executed on the peers, and peers maintain copies of the ledger. We

assigned one or multiple peers to each group.

Committee Peers:

Members of the committee are responsible for evaluating and validating every task and data during consensus. In a network with no sharding mechanism, we assume all peers in the blockchain network are responsible for validating and endorsing every executed task.

The clients can instantiate and submit different transactions on the network, as managed by the smart contracts. We have implemented four primary tasks of the DVS algorithm as smart contracts:

Initial grouping: In our prototype system, we assume that all admittance values are constant, and the system's topology will not change; therefore, this transaction will be executed once during the initialization of the network. The Impedance value of load buses, the Priority Index matrix, and the constant part of the Jacobian matrix are calculated through Matlab libraries, and their data are recorded for each group and combination of groups on the ledger. Note, however, that these values would change if the topology of the network changed.

The PI matrix has the same size as the admittance matrix without considering the reference bus, and the value of this matrix is the magnitude of the admittance value between each bus. The Impedance values are calculated for all load buses using Equation 4.3, and we also save the constant part of the Jacobian Matrix for calculating the sensitivity value. All these data are converted to the JSON format and sent through an HTTP request to a server. Then we call the InitialGrouping transaction and save all data of each group on the ledger.

ComputeVSI: This transaction receives the data of PMU measurements and uses the DVS algorithm to calculate the VSI for each load bus. Then the VSI values are sorted via the shell sort algorithm, and if the minimum VSI is less than the specified threshold, the LocalController transaction will be called. This transaction

also reads information from the ledger, such as the impedance values of the local group determined during the initial grouping. Algorithm 1 shows the pseudo-code of ComputeVSI.

Algorithm 1 ComputeVSI

```
1: Input: PMU measurements, GroupId
2: Impedance= Retrieve the data from the ledger(GroupId)
3: Calculate V_th and Z_th for each load bus (3)(4)
4: Calculate VSI for each load bus (5)
5: Sort VSI ascending
6: if VSI[0] ≤Threshold then
7:    Call LocalController
8: else
9:    if mergedgroup == true then
10:      Split the group
11:   end if
12: end if
```

LocalController: This transaction will be called inside of ComputeVSI to calculate the required reactive power to stabilize a weak bus, and activate VVCs to deliver it. The steps of this transaction are shown in Algorithm 2. It first retrieves the PI and Jacobian matrix from the ledger. Then it finds the electrically closest VVC to the weak bus and calculates the required reactive power. Two libraries (Jacobian, and List) were developed to help implement the DVS algorithm on smart contracts. The first one is used to calculate the Jacobian values with real-time voltage angle and magnitude, and the second one is used to find the top priority buses using the PI matrix. In the end, we save the amount of reactive power that is injected into the bus on the ledger. This data can be used to calculate the amount of money that we owe to each source, or predict the amount of power we may need in the grid in different periods, allowing us to actively prevent voltage collapse.

Jacobian Libarary: This class is used to calculate the sensitivity values; the constructor function receives the constant part of the Jacobian matrix, which is the magnitude and phase angle of admittance, along with voltages of each bus, and load and tie buses index value. This class will not calculate the whole matrix for all the

Algorithm 2 LocalController

```
1: Input: WeakBus, Resources, GroupId
2: PI, Jacobian= Retrieve the data from the ledger(GroupId)
3: w=WeakBus
4: Calculate δQ_w / δV_w
5: Calculate Q_req (6)
6: if resource available at WeakBus then
7:    Activate a VVC at WeakBus
8: else
9:    List= Create PI list
10:   while List is not Empty do
11:      i=List.GetNextPriority()
12:      Calculate δQ_i / δV_w
13:      Calculate Q_req (6)
14:      if Resource available at Bus i then
15:         Activate a VVC at Bus i
16:         Break
17:      end if
18:   end while
19: end if
20: Save VVC_index, Q_req,and GroupId on the ledger
```

buses, and it just calculates the values that we need during the LocalController. Two functions have been developed to calculate the sensitivity value: the getJacobianDiagonalValue(index) function receives the index of the bus and uses Equation 4.10 to compute the $\frac{\delta Q_{index}}{\delta V_{index}}$; the getJacobianValue(i, w, parent) function receives index of the weak bus (w), the bus that we want to calculate the sensitivity value respect to the weak bus (i), and the index of the bus that connects these two buses to each other (parent). If the two buses are directly connected to each other, the function uses Equation 4.11 to calculate the $\frac{\delta Q_j}{\delta V_w}$; otherwise, it uses the chain rule method (recursive function) shown in Equation 4.14. The Memoization technique in dynamic programming is used to optimize the process.

$$\frac{\delta Q_i}{\delta V_w} = getJacobianValue(i, parent, -1) * (1/getJacobianDiagonalValue(parent)) \; getJacobianValue(parent, w, -1).$$
$$\frac{\delta Q_i}{\delta V_w} = \frac{\delta Q_i}{\delta V_{parent}} * \frac{\delta V_{parent}}{\delta Q_{parent}} * \frac{\delta Q_{parent}}{\delta V_w}. \tag{4.14}$$

List Library: This class was developed to find the next priority bus in LocalController. This class has three functions: the addFirstRow(listofbuses) function receives the PI matrix that shows the connection of other buses to the weak bus and sorts the buses based on the electrical distance, which represents by the magnitude of admittance; the addToSortedList(node) is responsible for finding the position of each bus in a sorted list with the order complexity of O(log n); and the ReplaceFirstNode(listofbuses) function pops the first bus from sorted list and replaces it with its child, which uses the second function to add children to the sorted list.

GlobalController: Suppose the LocalController transaction cannot find enough power resources. In that case, the system will call this transaction to find an adjacent group with additional available resources, and send control actions to merge the data of these two groups. After that, the ComputeVSI and LocalController will be called on the merged data until the weak bus is stabilized. Once this is accomplished, the two groups will again be split. In merging the groups, we consider that one of the group aggregator devices is responsible for aggregating all PMUs of two groups.

4.5.2 Sharding Mechanism

In the blockchain network, every transaction must be authorized by all or the majority of nodes to ensure security. Due to this reason, it is challenging to improve scalability and transaction speed. Any increases in computing power cannot enhance the transaction speed; this is why the blockchain is not scalable in large networks. Sharding is one of the most practical solutions to achieve a scaling blockchain system,

where the calculation, storage, and processing can be conducted in parallel. Corbett et al. [148] proposed the sharding method in 2013, which is commonly used in distributed databases and cloud infrastructure. Sharding techniques and the scalability of blockchains have been investigated in various studies, and applications such as IoT networks [149], federated learning [150][151], and 6G networks [152]. Some of the studies also proposed a new blockchain protocol and cross-shard techniques such as RapidChain [38] and OmniLedger [153] or integrate sharding with permissioned and permissionless blockchain [154] [155].

Adding sharding to the blockchain-based DVS allows scalability by splitting the blockchain network into smaller subgroups or smaller subnetworks. In this way, each node only needs to process a small part of the work and transactions in different shards can be processed in parallel. The DVS is a real-time control algorithm and needs to react to each change in the power grid to prevent instability. However, as the size of the power grid controlled by DVS grows, it becomes likely that the many components of the grid are in constant flux. Hence, transaction volume and latency can be serious issues, which may delay and thus undermine DVS control actions in a rapidly-evolving voltage collapse. The sharding mechanism is designed to reduce storage and communication requirements while increasing throughput as the number of shards increases (while the physical network size remains constant) [156].

In this work, we employed a sharding technique to make our system more scalable and adoptable by expanding the network. In this technique, instead of using all peers to endorse and validate tasks in the system, we assign a smaller group of committee peers for each shard. Each shard can be responsible for validating shard-level transactions of one or multiple adjacent groups. Figure 4.2 provides an overview of our sharding model. Moreover, In this model all shards node hold a copy of the same ledger in the whole network, which prevents inconsistency in the network transaction records.

If we assign enough reactive sources for each load bus in a group, the chance of

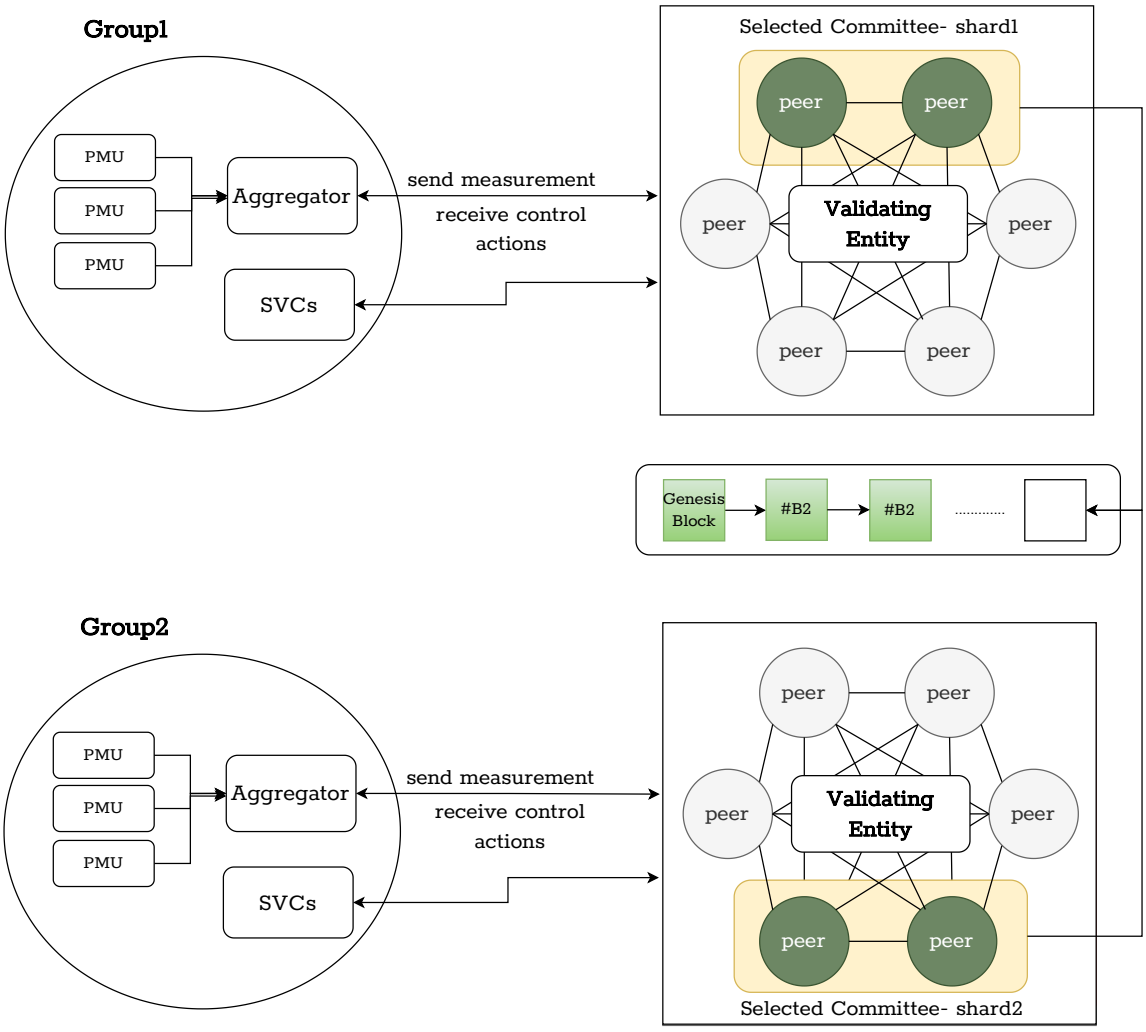

Figure 4.2: Blockchain-based model architecture for the DVS algorithm - sharding mechanism.

solving the voltage stability problem in each group will be increased. Therefore, we can split transactions based on the importance and occurrence frequency to shard level and mainchain level. Shard level refers to the transactions executed and endorsed by a subset of peers in each shard, and mainchain level refers to the transactions executed by all peers in the network. At the mainchain level, we assume that the global controller and all data related to the topology of each group will be executed and validated by all peers or a subset of peers assigned to every single group. This will help to preserve high security for critical control actions and access to the data of all groups without needing cross-shard communication. The other transactions such as ComputeVSI and LocalController, which occur within a single group, are executed in the shard level consensus.

4.6 Implementation and Deployment

In this section we discuss our simulation tools, experimental method, and evaluation results.

4.6.1 Simulation Tools

MATPOWER

In the power system operation, the power flow or load flow algorithm is used to compute voltages at different buses, line flows in the network, and system losses. The Optimal Power Flow (OPF) determines the best operating levels for electric power plants to meet demands given throughout a transmission network, usually to minimize operating costs. OPF is an extension of the problem of optimal economic dispatch (ED) of generation in traditional power systems introduced by Carpentier [157]. The OPF method is an instance of constrained optimization used to find the optimal state of any grid under system constraint conditions, such as loss minimization, reactive power limits, thermal limits of transmission lines, and reactive power optimization. The important feature of OPF is the presence of the load flow equations in the

set of equality constraints. Most OPF techniques are classified as traditional and metaheuristic based.

To simulate the power grid network and solve power flow and optimal power flow problems in this project, we used a package of Matlab called Matpower [2] [19]. Matpower includes the standard power flow or load flow solvers for both AC and DC power flow problems, which involves solving the set of voltages and flows in a network corresponding to a specified pattern of load and generation. For general AC power problems, Matpower introduced four different algorithms: The standard Newton's method [158], which is a default algorithm and uses polar form and a full Jacobian updated at each iteration. The Newton method uses nodal current balance equations or cartesian/hybrid representations for voltage [159]. The fast-decoupled method [160], specifically the XB and BX methods are described in [161]. The Standard Gauss-Seidel method comes from Glimm and Stagg [162].

Matpower has many options for selecting among the available solution algorithms, controlling the behaviour of the algorithms and determining the details of the pretty-printed output. The input data for the case to be simulated is referred to as a "Matpower case", which specify as a set of data matrices packaged as the fields of a Matlab struct denoted by the variable mpc.

In this project, we used the IEEE 30 bus case for our experiment, which is available in the data section of Matpower. The IEEE 30-bus test case represented a simple approximation of the American Electric Power system in December 1961 [131]. The system has 15 buses, 2 generators, and 3 synchronous condensers. The buses are either 132 or 33 kV, and the test case does not have line limits. Figure 4.3 represents the IEEE 30 bus system, which is split into three groups based on the DVS algorithm.

[2] https://matpower.org/

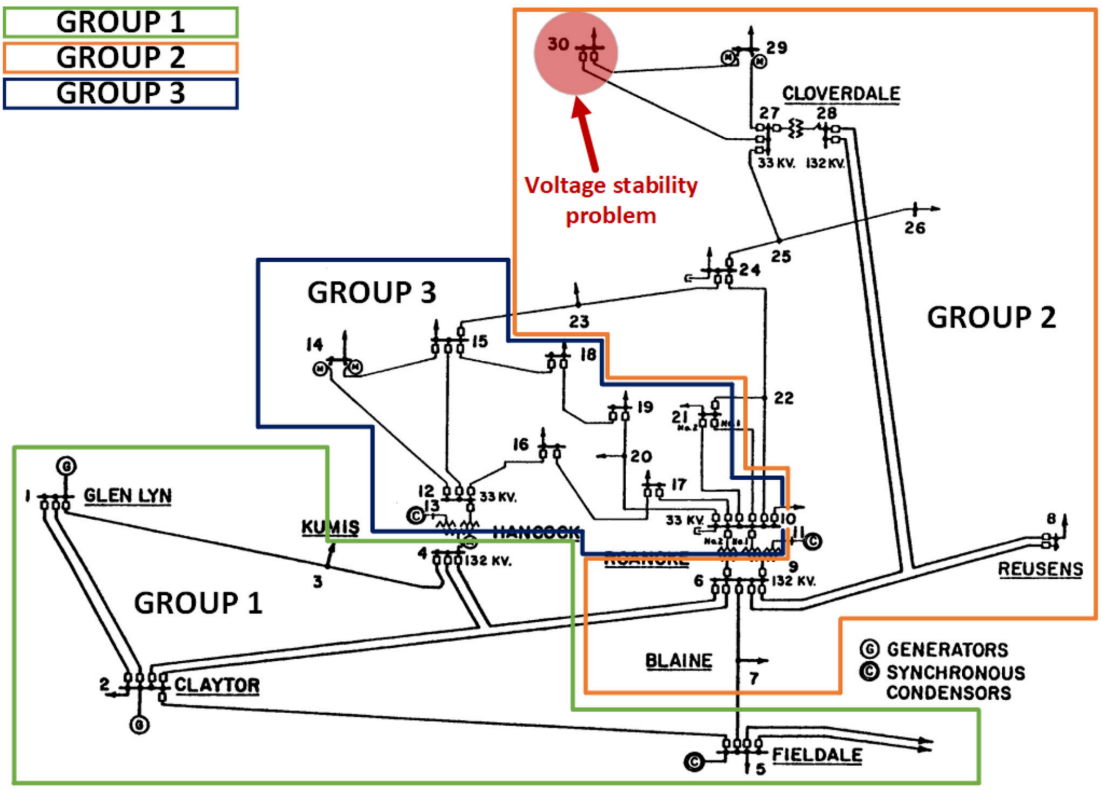

Figure 4.3: IEEE 30 bus system and grouping information represented by the DVS paper [16].

HyperLedger Fabric and Docker

We used Hyperledger Fabric [3] [18], a permissioned platform, to implement our models [4]. Fabric has a modular and configurable architecture and supports plug-n-play consensus and membership services. The execute-order-validate architecture for transactions provided by Fabric allows each peer to evaluate models in parallel within each shard, as opposed to most public blockchain platforms that have first-order transactions and execute them sequentially. Fabric also provides communication through channels that is a private layer of communication between two or more specific network members such as organizations and peers. Each transaction executed on a channel must be authenticated and authorized by channel members to transact on that channel. Peers can be members of multiple channels and perform channel related operations.

To handle all components and services of Hyperledger Fabric on a single machine, Fabric uses a Docker[5] platform to simulate the Blockchain. Docker is an open-source platform that separates applications from infrastructure and provides the ability to package and run an application in an isolated environment called a container. Docker, like a Virtual Machine (VM), provides an isolated environment for applications to run without interference from other apps. A Docker image is a template or instruction to create or instantiate a container. Hyperledger Fabric provides docker images for all components such as peer, CA, CouchDB, orderers, etc. It also uses Docker Compose [6] to help define and share multi-container application services.

HyperLedger Caliper

HyperLedger Caliper [20] is an open-source benchmark tool designed with scalability and extensibility to integrate with today's popular monitoring and infrastructure

[3]https://hyperledger-fabric.readthedocs.io/en/release-2.2/
[4]https://github.com/Scalable-Blockchain-Systems/DVSCode
[5]https://www.docker.com/
[6]https://docs.docker.com/compose/

Table 4.1: Experimental Configuration

Component	Version	CPU	GPU	RAM	Disk (SSD)
Caliper Benchmark	Caliper 0.4.2	Intel Core i7-9700K	GeForce RTX 2080 TI	62.8 GB	500 GB (SSD)
Fabric peer	Fabric 2.3.3	Intel Core i7-9700K	GeForce RTX 2080 TI	62.8 GB	500 GB (SSD)
Matpower	Matpower 7.1	Intel Core i7-9700K	GeForce RTX 2080 TI	62.8 GB	500 GB (SSD)

solutions for different blockchain platforms such as Hyperledger Besu, Ethereum, Hyperledger Fabric, and others. Caliper allows simulation of various workloads for a system to measure the performance of blockchain in terms of Throughput, Success rate and Latency with a set of predefined use cases. For this purpose, it generates a workload against a specific system under test (SUT) and continuously monitors its responses, then generates a report based on the observed SUT response.

To set up a Caliper we need to define three different files: first the Network Configuration file, which defines the network's topology and includes the description of different configuration elements such as organization, orderers, channels, Certificate Authority (CA), etc. Second, the Benchmark Configuration file determines the monitoring and observing settings to define what metric should be gathered and how the process gathers the information by selecting from different testing methods. Third, the Workload Configuration defines the logic pertaining to the business, benchmark or user behaviour. It is the brain of the SUT and decides which kind of transaction needs to be submitted at this moment.

Accordingly, Caliper includes two processes: a master process and numerous worker processes. The master process is responsible for initiating the SUT and schedules the configured round, and generates performance reports based on the transactions. The worker processes are the backbone of Caliper's scalability and are responsible for sending a transaction with a specified configuration independently of each other.

4.6.2 Experiments

For our experiments, we seek to quantify what benefit, if any, is provided to our prototype system by implementing sharding. Accordingly, we first measure throughput and latency without sharding, and then compare this with specific levels of sharding; specifically, a model with two shards and a model with three shards.

Networks run locally on a single machine simulating a Fabric test-network with a single orderer running Raft [35]. For the no-shard model, we consider six peers, each owned by different organizations and a certificate authority. Every group sends their requests to three different peers, and all peers are responsible for validating and endorsing a transaction. In the 2-shards model, we consider that each shard has three peers that evaluate models in parallel within each shard. For the 3-shards model, we assume that each shard has two peers responsible for validating each group's requests. We implement two smart contracts, one for ComputeVSI and local transactions called VSIContract, and the other for handling Global Controler called GlobalContract. The smart contract is known as a chaincode in Fabric and deployed to a specific channel. The channels are used to simulate shards in the system, where each channel operates independently with the ability to have different membership and endorsement policies. All six peers are a member of the "mainchain" channel that we deployed the GlobalContract smart contract on, and we deployed a VSIContract to each shard channel.

For the power grid simulation, we applied the DVS algorithm to the IEEE 30 bus network [163] (also used in [16]). Following [16], we split the grid into three different groups and save the admittance, PI matrix, and constant part of Jacobian for each group on the ledger. We begin with a normal scenario, with no voltage stability problems. We collected this normal data for testing a ComputeVSI transaction with Hyperledger Caliper. Then, we increase a load of some of the buses, decreasing the stability margin to trigger the LocalController transaction.

To find an estimation for the threshold of voltage stability index for each group and separate the normal and unstable data, we conducted several experiments on different load buses. We multiplied the load by 22, 2, or 4 based on the sensitivity of each bus; then, we applied a DVS controller algorithm and injected estimated reactive power into each bus to stabilize the load buses. It takes multiple steps and injections for some buses to become stable and the voltage to increase. During the experiment, we recorded all the VSIs and compared them with the original value of the IEEE 30 case system. The results and defined scenarios are available in Appendix A.

For evaluation, we used Hyperledger Caliper [20] in which an independent process is responsible for sending a transaction with a specified configuration, and the system's performance (latency, throughput, success rate) is reported. We conducted tests with varying numbers of workers, transaction numbers, and send rates (TPS) to evaluate the limits of each model for ComputeVSI and LocalController transactions. The experimental configuration is summarized in Table 4.1.

4.6.3 Results

The ComputeVSI transaction is the only transaction that will be called multiple times every second (based on the measurement frequency of the PMUs). In contrast, the Control transaction will be called a few times when needed. Therefore, most of the workload is run with the normal data, invoking just the ComputeVSI transaction.

To test the maximum throughput achieved by our system, we measured the send rate against the system throughput and average latency in one experiment with normal data and in another experiment with unstable data. Figure 4.4 shows the result for normal data that just ran the ComputeVSI transaction, and Figure 4.5 shows the result for unstable data that ran both the ComputeVSI and LocalController transactions. These workloads are run with 3 Caliper workers over 8000 transactions. By increasing the send rate, each model will reach a point that becomes saturated and is unable to handle a higher send rate. As shown in the figures, the no-shard model

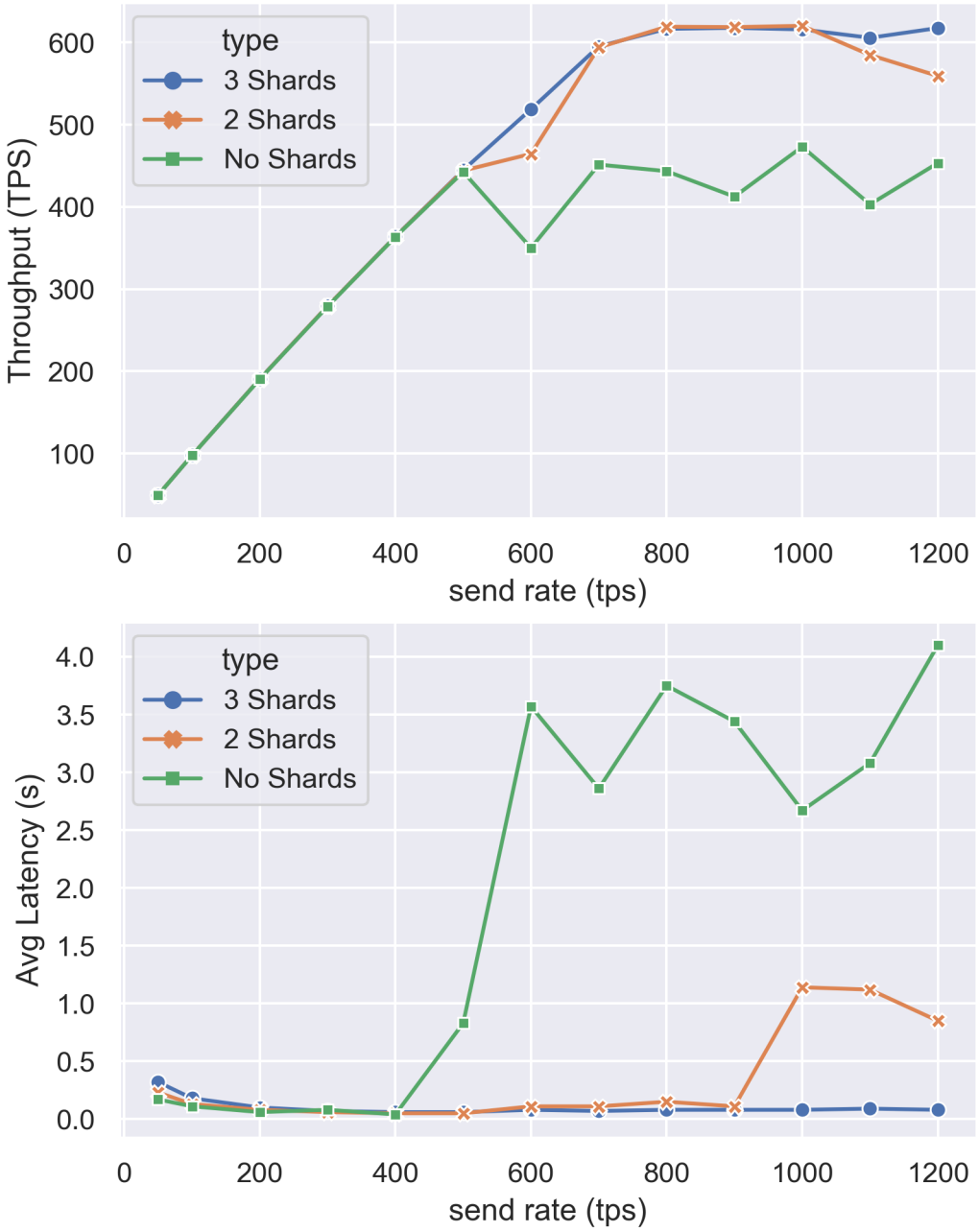

Figure 4.4: Send rate vs. system throughput (TPS) & Average response latency - Fixed values: 3 worker over 8000 transactions - ComputeVSI transaction.

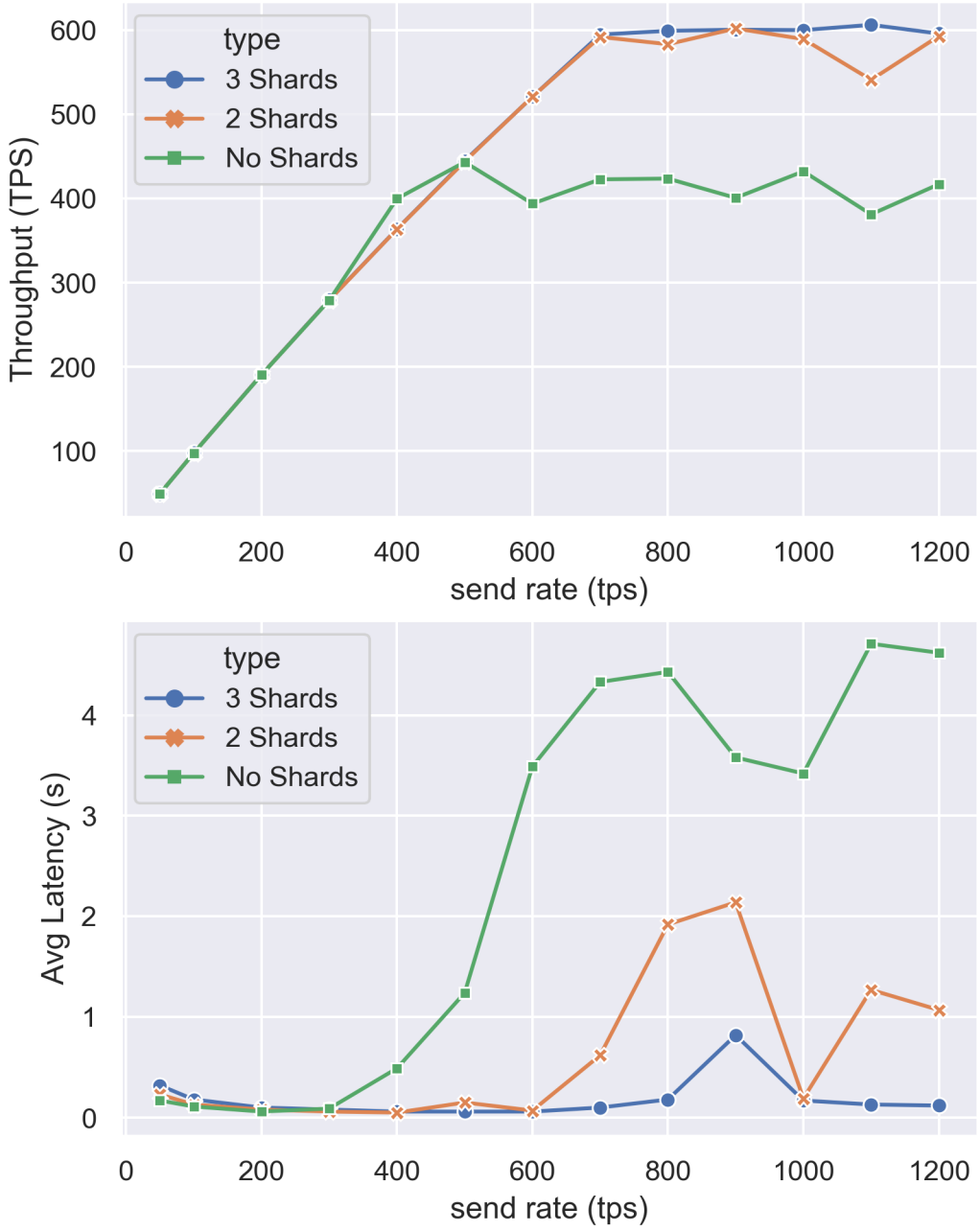

Figure 4.5: Send rate vs. system throughput (TPS) & Average response latency - Fixed values: 3 worker over 8000 transactions - ComputeVSI+LocalController transactions.

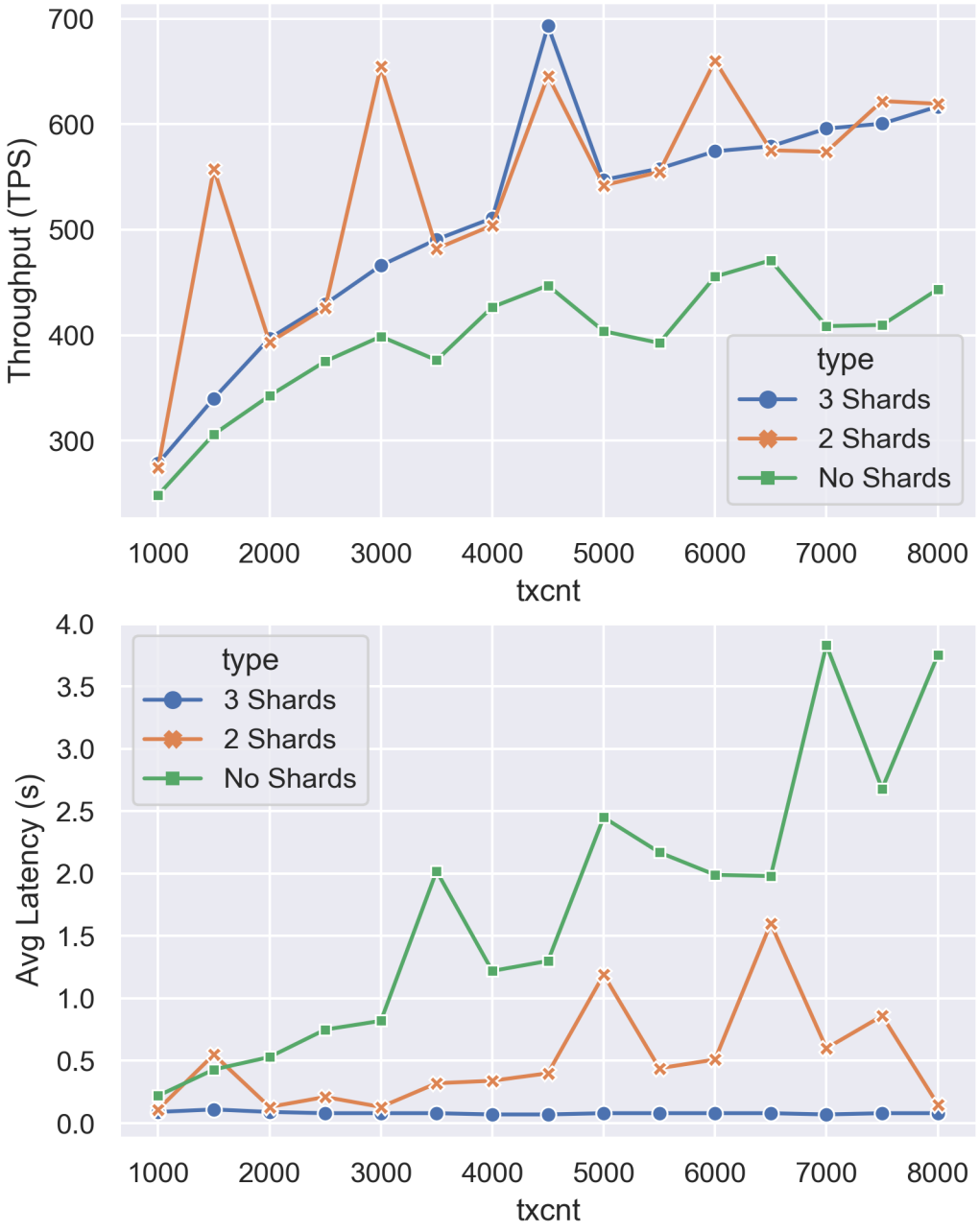

Figure 4.6: Transaction number (txcnt) vs. System throughput (TPS) & Average response latency - Fixed values: 3 worker with tps of 800 - ComputeVSI transaction.

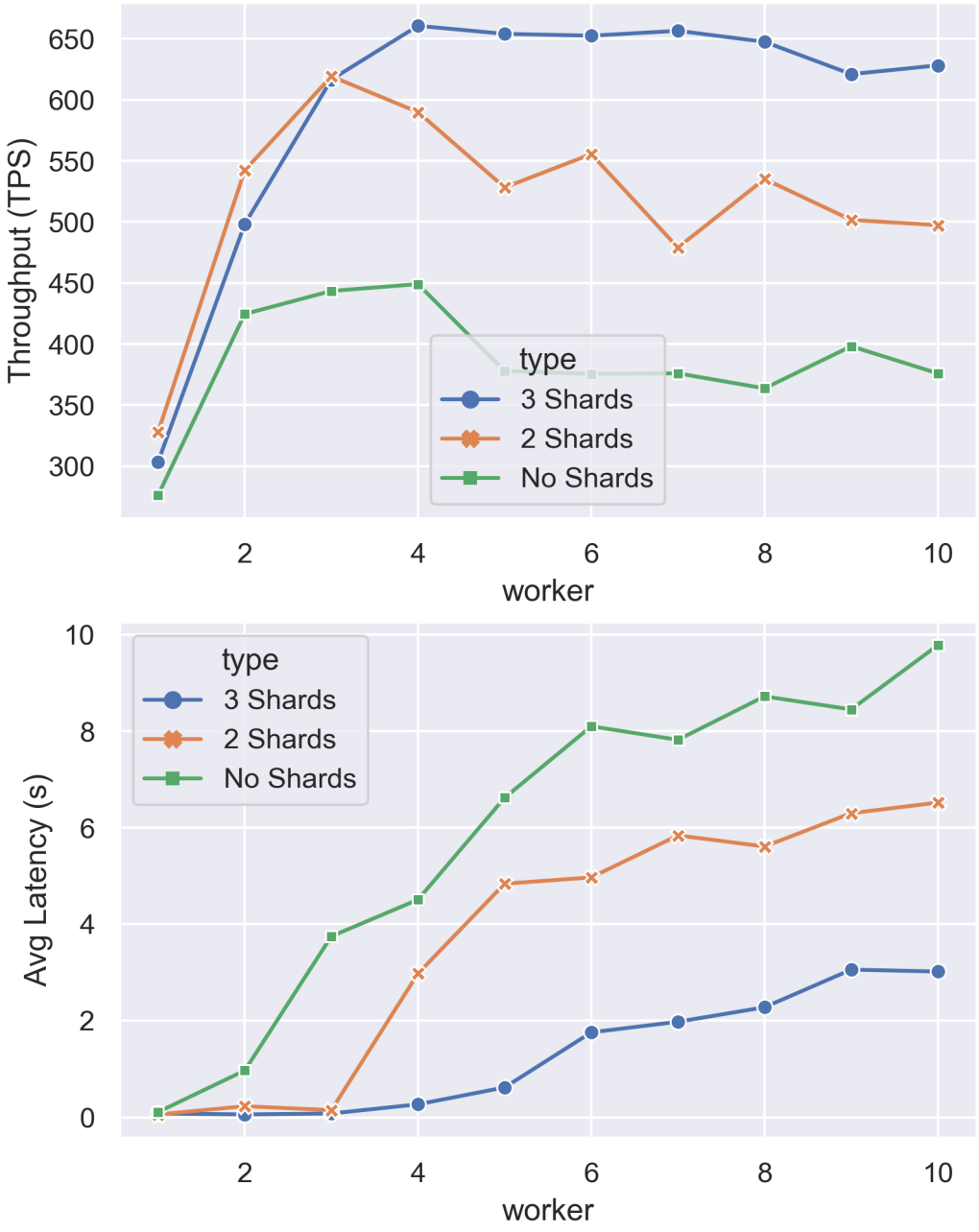

Figure 4.7: Number of workers vs. System throughput (TPS) & Average response latency - Fixed values: 8000 transaction with tps of 800 - ComputeVSI transaction.

gets to this threshold sooner than the sharding model.

To explore the limits of a usage surge, we tested the number of transactions sent by the system with respect to throughput and average latency. This workload is run with 3 caliper workers and a fixed sent rate of 800 tps. Figure 4.6 shows the throughput and latency of this workload. As can be seen, the result shows that the sharding model can significantly improve the overall throughput and latency.

Finally, we tested how the system handles concurrent requests by running multiple workloads, varying the number of caliper workers. This workload configuration sends 8000 transactions with a send rate of 800 to measure the system throughput and average latency. The number of caliper workers allows us to scale the workload generation, and each worker processes performs the actual workload generation in parallel and independently of each other. As we can see in Figure 4.7, the throughput of the system has a general downward trend in the system throughput with respect to the number of workers. Similarly, we can see an upward trend in average latency. We can see that the number of shards plays the most important role for average latency due to these workloads being able to operate in parallel across shards.

4.7 Conclusion

In this work, we designed and implemented a scalable blockchain-based DVS system to compute a voltage stability index and take control actions in the electrical power system. Our smart contracts regulate automatic computations of decentralized DVS algorithm and manage resources in the electric power system. The experimental results demonstrate that permissioned blockchains can handle a large number of power grid transactions in a few seconds, and we can improve system performance close to linearly with the addition of shards. This helps address the scalability issue related to blockchain consensus in a large-scale power grid network.

Chapter 5

Conclusions and Future Work

In this work, we first conducted a systematic review of the design and performance of blockchain-based smart contract systems in Smart Grid applications, and investigated soloutions to the data storage and Interoperability concerns. We then proposed a blockchain-based smart contract system for decentralized control and monitoring in energy systems, and we enhanced the performance and scalability of the system by applying a sharding mechanism. The following summarizes the findings and results of each of our studies as well as possible future research directions.

In Chapter 3 we presented a systematic literature review on the design of smart contracts in a blockchain-based energy system. We investigated 62 papers in energy systems concerning blockchain fundamentals and smart contracts for energy systems and their interaction with the technical elements of the Smart Grid. Based on the purpose of smart contracts deployed in energy systems and the Smart Grid application domain, we categorized the studies into market operations, ancillary services, auditing and monitoring, and cybersecurity. By examining different approaches, we identified two different limitations of the blockchain platforms: data storage and the scalability problem. We proposed solutions to the challanges based on existing research to solve each problem, such as a removable ledger, interoperable blockchains, InterPlanetary File System (IPFS), etc.

In Chapter 4, we proposed a scalable model for decentralized voltage stability

control using blockchain and smart contract technology to analyze and investigate the performance of blockchain as a real-time application in a Smart Grid. Our model used smart contracts to communicate and manage devices and automate the monitoring and control algorithm in real-time. To scale our model up to a larger network, we applied sharding techniques and evaluated our model with a different number of shards. We showed that sharding can improve linearly the latency and throughput of the system. Using blockchain as a decentralized platform for real-time monitoring and control also provides a platform to analyze the trace and pattern of data associated with voltage collapse. Thus, blockchain can be used to identify potential threats in the system, predict possible vulnerabilities in the power system, and integrate automatic preventative actions. This will also provide data on the required number of reactive power sources that need to be added to improve the grid's stability. Hence, in future work, it is worth investigating and integrating with our model the Machine Learning models and AI techniques to reduce the complexity or give better intuition about the system.

Bibliography

[1] K. Honari, X. Zhou, S. Rouhani, S. Dick, H. Liang, Y. Li, and J. Miller, "A scalable blockchain-based smart contract model for decentralized voltage stability using sharding technique," *arXiv preprint arXiv: arXiv:2206.13776*, 2022.

[2] W. Anjali, A. Josef, and R. Marylène, "Smart grid in canada 2018," 2019–066 RP–FIN DER–SGNETS, Natural Resources Canada.

[3] Dublin. (Aug. 21, 2022). The worldwide smart grid technology industry is projected to reach $117 billion by 2027, [Online]. Available: https://www.globenewswire.com/news-release/2022/06/21/2466120/0/en/The-Worldwide-Smart-Grid-Technology-Industry-is-Projected-to-Reach-117-Billion-by-2027.html (visited on 07/12/2022).

[4] P. Fox. (Nov. 18, 2018). $1.5 to $2 trillion investment needed in U.S. electric utility industry by 2030, [Online]. Available: https://www.smart-energy.com/regional-news/north-america/1-5-to-2-trillion-investment-needed-in-u-s-electric-utility-industry-by-2030/ (visited on 07/12/2022).

[5] (Jan. 2022). United States Department of Energy- 2020 smart grid system report, [Online]. Available: https://www.energy.gov/sites/default/files/2022-05/2020%20Smart%20Grid%20System%20Report_0.pdf (visited on 07/12/2022).

[6] IEA. (2022). Canada 2022, IEA, Paris, [Online]. Available: https://www.iea.org/reports/canada-2022 (visited on 07/12/2022).

[7] K. Solaun and E. Cerdá, "Climate change impacts on renewable energy generation. a review of quantitative projections," *Renewable and sustainable energy Reviews*, vol. 116, p. 109 415, 2019.

[8] L.-N. Guo, C. She, D.-B. Kong, S.-L. Yan, Y.-P. Xu, M. Khayatnezhad, and F. Gholinia, "Prediction of the effects of climate change on hydroelectric generation, electricity demand, and emissions of greenhouse gases under climatic scenarios and optimized ann model," *Energy Reports*, vol. 7, pp. 5431–5445, 2021.

[9] N. O. Kapustin and D. A. Grushevenko, "Long-term electric vehicles outlook and their potential impact on electric grid," *Energy Policy*, vol. 137, p. 111 103, 2020.

[10] X. Wang, W. Yang, S. Noor, C. Chen, M. Guo, and K. H. van Dam, "Blockchain-based smart contract for energy demand management," *Energy Procedia*, vol. 158, pp. 2719–2724, 2019.

[11] S. Rouhani and R. Deters, "Security, performance, and applications of smart contracts: A systematic survey," *IEEE Access*, vol. 7, pp. 50 759–50 779, 2019.

[12] M. L. Di Silvestre, P. Gallo, M. G. Ippolito, R. Musca, E. R. Sanseverino, Q. T. T. Tran, and G. Zizzo, "Ancillary services in the energy blockchain for microgrids," *IEEE Transactions on Industry Applications*, vol. 55, no. 6, pp. 7310–7319, 2019.

[13] P. Danzi, M. Angjelichinoski, Č. Stefanović, and P. Popovski, "Distributed proportional-fairness control in microgrids via blockchain smart contracts," in *2017 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, IEEE, 2017, pp. 45–51.

[14] M. Baza, M. Nabil, M. Ismail, M. Mahmoud, E. Serpedin, and M. A. Rahman, "Blockchain-based charging coordination mechanism for smart grid energy storage units," in *2019 IEEE International Conference on Blockchain*, IEEE, 2019, pp. 504–509.

[15] X. Yang, G. Wang, H. He, J. Lu, and Y. Zhang, "Automated demand response framework in ELNs: Decentralized scheduling and smart contract," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 50, no. 1, pp. 58–72, 2019.

[16] H. Lee, A. K. Srivastava, V. V. Krishnan, S. Niddodi, and D. E. Bakken, "Decentralized voltage stability monitoring and control with distributed computing coordination," *IEEE Systems Journal*, 2021.

[17] H. Lee, S. Niddodi, A. Srivastava, and D. Bakken, "Decentralized voltage stability monitoring and control in the smart grid using distributed computing architecture," in *2016 IEEE Industry Applications Society Annual Meeting*, IEEE, 2016, pp. 1–9.

[18] E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. De Caro, D. Enyeart, C. Ferris, G. Laventman, Y. Manevich, *et al.*, "Hyperledger fabric: A distributed operating system for permissioned blockchains," in *Proceedings of the Thirteenth EuroSys Conference*, 2018, pp. 1–15.

[19] R. D. Zimmerman and C. E. Murillo-Sánchez, "Matpower 6.0 user's manual," *Power Systems Engineering Research Center*, vol. 9, 2016.

[20] (Mar. 2022). Hyperledger caliper, [Online]. Available: https://www.hyperledger.org/use/caliper (visited on 03/12/2022).

[21] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," *Decentralized Business Review*, p. 21 260, 2008.

[22] T. Haerder and A. Reuter, "Principles of transaction-oriented database recovery," *ACM Computing Surveys (CSUR)*, vol. 15, no. 4, pp. 287–317, 1983.

[23] D. Pritchett, "Base: An acid alternative: In partitioned databases, trading some consistency for availability can lead to dramatic improvements in scalability.," *Queue*, vol. 6, no. 3, pp. 48–55, 2008.

[24] S. Tai, J. Eberhardt, and M. Klems, "Not acid, not base, but salt," in *Proceedings of the 7th International Conference on Cloud Computing and Services Science*, 2017, pp. 755–764.

[25] V. Buterin *et al.*, "A next-generation smart contract and decentralized application platform," *white paper*, vol. 3, no. 37, 2014.

[26] R. G. Brown, J. Carlyle, I. Grigg, and M. Hearn, "Corda: An introduction," *R3 CEV, August*, vol. 1, p. 15, 2016.

[27] N. Szabo, "Formalizing and securing relationships on public networks," *First Monday*, 1997.

[28] C. Dannen, *Introducing Ethereum and solidity*. Springer, 2017, vol. 318.

[29] V. Buterin, G Wood, V Zamfir, and J Coleman, "Notes on scalable blockchain protocols," *Ethereum Foundation*, vol. 31, 2015.

[30] Y. Huang, Y. Bian, R. Li, J. L. Zhao, and P. Shi, "Smart contract security: A software lifecycle perspective," *IEEE Access*, vol. 7, pp. 150 184–150 202, 2019.

[31] G.-T. Nguyen and K. Kim, "A survey about consensus algorithms used in blockchain," *Journal of Information Processing Systems*, vol. 14, no. 1, pp. 101–128, 2018.

[32] J. Garay, A. Kiayias, and N. Leonardos, "The bitcoin backbone protocol: Analysis and applications," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, Springer, 2015, pp. 281–310.

[33] (Jun. 2017). Fabian schuh and daniel larimer. bitshares 2.0: General overview. accessed june-2017., [Online]. Available: http://docs.bitshares.org/downloads/bitshares-general (visited on 06/12/2022).

[34] (Jun. 2018). Nem technical reference, 2018., [Online]. Available: https://nem.io/wp-content/themes/nem/files/NEMtechRef.pdf (visited on 06/30/2022).

[35] D. Ongaro and J. Ousterhout, "In search of an understandable consensus algorithm," in *2014 USENIX Annual Technical Conference (Usenix ATC 14)*, 2014, pp. 305–319.

[36] M. Castro and B. Liskov, "Practical byzantine fault tolerance and proactive recovery," *ACM Transactions on Computer Systems (TOCS)*, vol. 20, no. 4, pp. 398–461, 2002.

[37] (Jun. 2017). Istanbul byzantine fault tolerant consensus protocol, [Online]. Available: https://github.com/ethereum/EIPs/issues/650 (visited on 06/30/2022).

[38] M. Zamani, M. Movahedi, and M. Raykova, "Rapidchain: Scaling blockchain via full sharding," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, 2018, pp. 931–948.

[39] D. Yu, H. Xu, L. Zhang, B. Cao, and M. A. Imran, "Security analysis of sharding in the blockchain system," in *2021 IEEE 32nd Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, IEEE, 2021, pp. 1030–1035.

[40] Y. Dodis and A. Yampolskiy, "A verifiable random function with short proofs and keys," in *International Workshop on Public Key Cryptography*, Springer, 2005, pp. 416–431.

[41] M. H. Manshaei, M. Jadliwala, A. Maiti, and M. Fooladgar, "A game-theoretic analysis of shard-based permissionless blockchains," *IEEE Access*, vol. 6, pp. 78 100–78 112, 2018.

[42] J. Yun, Y. Goh, and J.-M. Chung, "Trust-based shard distribution scheme for fault-tolerant shard blockchain networks," *IEEE Access*, vol. 7, pp. 135 164–135 175, 2019.

[43] H. Liang, A. K. Tamang, W. Zhuang, and X. S. Shen, "Stochastic information management in smart grid," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 3, pp. 1746–1770, 2014.

[44] C. Greer, D. A. Wollman, D. E. Prochaska, P. A. Boynton, J. A. Mazer, C. T. Nguyen, G. J. FitzPatrick, T. L. Nelson, G. H. Koepke, A. R. Hefner Jr, *et al.*, "NIST framework and roadmap for smart grid interoperability standards, release 3.0," 2014.

[45] M. A. Gilani, A. Kazemi, and M. Ghasemi, "Distribution system resilience enhancement by microgrid formation considering distributed energy resources," *Energy*, vol. 191, p. 116 442, 2020.

[46] Y. Lin, Z. Bie, and A. Qiu, "A review of key strategies in realizing power system resilience," *Global Energy Interconnection*, vol. 1, no. 1, pp. 70–78, 2018.

[47] A. Hahn, R. Singh, C.-C. Liu, and S. Chen, "Smart contract-based campus demonstration of decentralized transactive energy auctions," in *2017 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT)*, IEEE, 2017, pp. 1–5.

[48] A. Kumari, A. Shukla, R. Gupta, S. Tanwar, S. Tyagi, and N. Kumar, "Et-deal: A p2p smart contract-based secure energy trading scheme for smart grid systems," in *IEEE INFOCOM 2020-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, IEEE, 2020, pp. 1051–1056.

[49] D. Han, C. Zhang, J. Ping, and Z. Yan, "Smart contract architecture for decentralized energy trading and management based on blockchains," *Energy*, vol. 199, p. 117 417, 2020.

[50] J. G. Monroe, P. Hansen, M. Sorell, and E. Z. Berglund, "Agent-based model of a blockchain enabled peer-to-peer energy market: Application for a neighborhood trial in perth, australia," *Smart Cities*, vol. 3, no. 3, pp. 1072–1099, 2020.

[51] M. Andoni, V. Robu, D. Flynn, S. Abram, D. Geach, D. Jenkins, P. McCallum, and A. Peacock, "Blockchain technology in the energy sector: A systematic review of challenges and opportunities," *Renewable and Sustainable Energy Reviews*, vol. 100, pp. 143–174, 2019.

[52] P. Zhuang, T. Zamir, and H. Liang, "Blockchain for cybersecurity in smart grid: A comprehensive survey," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 1, pp. 3–19, 2020.

[53] Y. Zhang, T. Huang, and E. F. Bompard, "Big data analytics in smart grids: A review," *Energy Informatics*, vol. 1, no. 1, pp. 1–24, 2018.

[54] T. Alladi, V. Chamola, J. J. Rodrigues, and S. A. Kozlov, "Blockchain in smart grids: A review on different use cases," *Sensors*, vol. 19, no. 22, p. 4862, 2019.

[55] S. S. Ali and B. J. Choi, "State-of-the-art artificial intelligence techniques for distributed smart grids: A review," *Electronics*, vol. 9, no. 6, p. 1030, 2020.

[56] S. Kushch and F. P. Castrillo, "A review of the applications of the block-chain technology in smart devices and dis-tributed renewable energy grids," *ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal*, vol. 6, no. 3, p. 75, 2017.

[57] J. Bao, D. He, M. Luo, and K.-K. R. Choo, "A survey of blockchain applications in the energy sector," *IEEE Systems Journal*, 2020.

[58] J. Abdella and K. Shuaib, "Peer to peer distributed energy trading in smart grids: A survey," *Energies*, vol. 11, no. 6, p. 1560, 2018.

[59] M. B. Mollah, J. Zhao, D. Niyato, K.-Y. Lam, X. Zhang, A. M. Ghias, L. H. Koh, and L. Yang, "Blockchain for future smart grid: A comprehensive survey," *IEEE Internet of Things Journal*, vol. 8, no. 1, pp. 18–43, 2020.

[60] J. Lu, S. Wu, H. Cheng, and Z. Xiang, "Smart contract for distributed energy trading in virtual power plants based on blockchain," *Computational Intelligence*, vol. 37, no. 3, pp. 1445–1455, 2021.

[61] E. Mengelkamp, B. Notheisen, C. Beer, D. Dauer, and C. Weinhardt, "A blockchain-based smart grid: Towards sustainable local energy markets," *Computer Science-Research and Development*, vol. 33, no. 1, pp. 207–214, 2018.

[62] S. Suthar and N. M. Pindoriya, "Blockchain and smart contract based decentralized energy trading platform," in *2020 21st National Power Systems Conference (NPSC)*, IEEE, 2020, pp. 1–5.

[63] Y. Amanbek, Y. Tabarak, H. K. Nunna, and S. Doolla, "Decentralized transactive energy management system for distribution systems with prosumer microgrids," in *2018 19th International Carpathian Control Conference (ICCC)*, IEEE, 2018, pp. 553–558.

[64] M. Sabounchi and J. Wei, "Towards resilient networked microgrids: Blockchain-enabled peer-to-peer electricity trading mechanism," in *2017 IEEE Conference on Energy Internet and Energy System Integration (EI2)*, IEEE, 2017, pp. 1–5.

[65] S. Myung and J.-H. Lee, "Ethereum smart contract-based automated power trading algorithm in a microgrid environment," *The Journal of Supercomputing*, vol. 76, no. 7, pp. 4904–4914, 2020.

[66] I. Dimobi, M. Pipattanasomporn, and S. Rahman, "A transactive grid with microgrids using blockchain for the energy internet," in *2020 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT)*, IEEE, 2020, pp. 1–5.

[67] K. Heck, E. Mengelkamp, and C. Weinhardt, "Blockchain-based local energy markets: Decentralized trading on single-board computers," *Energy Systems*, pp. 1–16, 2020.

[68] K.-L. Brousmichc, A. Anoaica, O. Dib, T. Abdellatif, and G. Deleuze, "Blockchain energy market place evaluation: An agent-based approach," in *2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, IEEE, 2018, pp. 321–327.

[69] S. Seven, G. Yao, A. Soran, A. Onen, and S. Muyeen, "Peer-to-peer energy trading in virtual power plant based on blockchain smart contracts," *IEEE Access*, vol. 8, pp. 175 713–175 726, 2020.

[70] S. Saxena, H. Farag, A. Brookson, H. Turesson, and H. Kim, "Design and field implementation of blockchain based renewable energy trading in residential communities," in *2019 2nd International Conference on Smart Grid and Renewable Energy (SGRE)*, IEEE, 2019, pp. 1–6.

[71] A. Muzumdar, C. Modi, G. Madhu, and C Vyjayanthi, "A trustworthy and incentivized smart grid energy trading framework using distributed ledger and smart contracts," *Journal of Network and Computer Applications*, vol. 183, p. 103 074, 2021.

[72] I. El-Sayed, K. Khan, X. Dominguez, and P. Arboleya, "A real pilot-platform implementation for blockchain-based peer-to-peer energy trading," in *2020 IEEE Power & Energy Society General Meeting (PESGM)*, IEEE, 2020, pp. 1–5.

[73] O. Bouachir, M. Aloqaily, Ö. Özkasap, and F. Ali, "Federatedgrids: Federated learning and blockchain-assisted p2p energy sharing," *IEEE Transactions on Green Communications and Networking*, 2022.

[74] E. Münsing, J. Mather, and S. Moura, "Blockchains for decentralized optimization of energy resources in microgrid networks," in *2017 IEEE conference on control technology and applications (CCTA)*, IEEE, 2017, pp. 2164–2171.

[75] Y. Wang, Z. Su, N. Zhang, J. Chen, X. Sun, Z. Ye, and Z. Zhou, "Spds: A secure and auditable private data sharing scheme for smart grid based on blockchain and smart contract," *IEEE Transactions on Industrial Informatics*, 2020.

[76] A. Dorri, F. Luo, S. S. Kanhere, R. Jurdak, and Z. Y. Dong, "Spb: A secure private blockchain-based solution for distributed energy trading," *IEEE Communications Magazine*, vol. 57, no. 7, pp. 120–126, 2019.

[77] I. Kounelis, G. Steri, R. Giuliani, D. Geneiatakis, R. Neisse, and I. Nai-Fovino, "Fostering consumers' energy market through smart contracts," in *2017 International Conference in Energy and Sustainability in Small Developing Economies (ES2DE)*, IEEE, 2017, pp. 1–6.

[78] H. You, H. Hua, and J. Cao, "A smart contract-based energy trading strategy in energy internet," in *2019 IEEE International Conference on Energy Internet (ICEI)*, IEEE, 2019, pp. 478–483.

[79] M. Afzal, Q. Huang, W. Amin, K. Umer, A. Raza, and M. Naeem, "Blockchain enabled distributed demand side management in community energy system with smart homes," *IEEE Access*, vol. 8, pp. 37 428–37 439, 2020.

[80] R. Khalid, N. Javaid, A. Almogren, M. U. Javed, S. Javaid, and M. Zuair, "A blockchain-based load balancing in decentralized hybrid p2p energy trading market in smart grid," *IEEE Access*, vol. 8, pp. 47 047–47 062, 2020.

[81] F. Knirsch, A. Unterweger, G. Eibl, and D. Engel, "Privacy-preserving smart grid tariff decisions with blockchain-based smart contracts," in *Sustainable Cloud and Energy Services*, Springer, 2018, pp. 85–116.

[82] K. Nakayama, R. Moslemi, and R. Sharma, "Transactive energy management with blockchain smart contracts for p2p multi-settlement markets," in *2019 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT)*, IEEE, 2019, pp. 1–5.

[83] Y. Li, R. Rahmani, N. Fouassier, P. Stenlund, and K. Ouyang, "A blockchain-based architecture for stable and trustworthy smart grid," *Procedia Computer Science*, vol. 155, pp. 410–416, 2019.

[84] E. S. Kang, S. J. Pee, J. G. Song, and J. W. Jang, "A blockchain-based energy trading platform for smart homes in a microgrid," in *2018 3rd International Conference on Computer and Communication Systems (ICCCS)*, IEEE, 2018, pp. 472–476.

[85] X. Zhang and M. Fan, "Blockchain-based secure equipment diagnosis mechanism of smart grid," *IEEE Access*, vol. 6, pp. 66 165–66 177, 2018.

[86] M. Utz, S. Albrecht, T. Zoerner, and J. Strüker, "Blockchain-based management of shared energy assets using a smart contract ecosystem," in *International Conference on Business Information Systems*, Springer, 2018, pp. 217–222.

[87] W Hu, Y. Hu, W. Yao, W. Lu, H. Li, and Z. Lv, "A blockchain-based smart contract trading mechanism for energy power supply and demand network," *Advances in Production Engineering & Management*, vol. 14, no. 3, pp. 284–296, 2019.

[88] L. Thomas, Y. Zhou, C. Long, J. Wu, and N. Jenkins, "A general form of smart contract for decentralized energy systems management," *Nature Energy*, vol. 4, no. 2, pp. 140–149, 2019.

[89] C. Liu, K. K. Chai, X. Zhang, E. T. Lau, and Y. Chen, "Adaptive blockchain-based electric vehicle participation scheme in smart grid platform," *IEEE Access*, vol. 6, pp. 25 657–25 665, 2018.

[90] N. Mhaisen, N. Fetais, and A. Massoud, "Secure smart contract-enabled control of battery energy storage systems against cyber-attacks," *Alexandria Engineering Journal*, vol. 58, no. 4, pp. 1291–1300, 2019.

[91] J. Gao, K. O. Asamoah, E. B. Sifah, A. Smahi, Q. Xia, H. Xia, X. Zhang, and G. Dong, "Gridmonitoring: Secured sovereign blockchain based monitoring on smart grid," *IEEE Access*, vol. 6, pp. 9917–9925, 2018.

[92] Y. Li, W. Yang, P. He, C. Chen, and X. Wang, "Design and management of a distributed hybrid energy system through smart contract and blockchain," *Applied Energy*, vol. 248, pp. 390–405, 2019.

[93] Y. Yang, M. Liu, Q. Zhou, H. Zhou, and R. Wang, "A blockchain based data monitoring and sharing approach for smart grids," *IEEE Access*, 2019.

[94] K. Gai, Y. Wu, L. Zhu, M. Qiu, and M. Shen, "Privacy-preserving energy trading using consortium blockchain in smart grid," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 6, pp. 3548–3558, 2019.

[95] K. Gai, Y. Wu, L. Zhu, L. Xu, and Y. Zhang, "Permissioned blockchain and edge computing empowered privacy-preserving smart grid networks," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 7992–8004, 2019.

[96] F. Lombardi, L. Aniello, S. De Angelis, A. Margheri, and V. Sassone, "A blockchain-based infrastructure for reliable and cost-effective IoT-aided smart grids," 2018.

[97] A. Dorri, F. Luo, S. Karumba, S. Kanhere, R. Jurdak, and Z. Y. Dong, "Temporary immutability: A removable blockchain solution for prosumer-side energy trading," *Journal of Network and Computer Applications*, vol. 180, p. 103 018, 2021.

[98] J. Wang, L. Wu, K.-K. R. Choo, and D. He, "Blockchain-based anonymous authentication with key management for smart grid edge computing infrastructure," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 3, pp. 1984–1992, 2019.

[99] W. Wang, H. Huang, L. Zhang, and C. Su, "Secure and efficient mutual authentication protocol for smart grid under blockchain," *Peer-to-Peer Networking and Applications*, vol. 14, no. 5, pp. 2681–2693, 2021.

[100] D. Zheng, K. Deng, Y. Zhang, J. Zhao, X. Zheng, and X. Ma, "Smart grid power trading based on consortium blockchain in Internet of Things," in *International Conference on Algorithms and Architectures for Parallel Processing*, Springer, 2018, pp. 453–459.

[101] Y. N. Aung and T. Tantidham, "Ethereum-based emergency service for smart home system: Smart contract implementation," in *2019 21st International Conference on Advanced Communication Technology (ICACT)*, IEEE, 2019, pp. 147–152.

[102] D. Thomas, M. Iain, and G. Dale, "Virtual power plants leveraging energy flexibility in regional markets," *CIRED-Open Access Proceedings Journal*, vol. 2017, no. 1, pp. 2939–2943, 2017.

[103] Q. Wang, X. Zhu, Y. Ni, L. Gu, and H. Zhu, "Blockchain for the IoT and industrial IoT: A review," *Internet of Things*, vol. 10, p. 100 081, 2020.

[104] E. Mengelkamp, J. Gärttner, K. Rock, S. Kessler, L. Orsini, and C. Weinhardt, "Designing microgrid energy markets: A case study: The brooklyn microgrid," *Applied Energy*, vol. 210, pp. 870–880, 2018.

[105] H. A. Khattak, K. Tehreem, A. Almogren, Z. Ameer, I. U. Din, and M. Adnan, "Dynamic pricing in industrial internet of things: Blockchain application for energy management in smart cities," *Journal of Information Security and Applications*, vol. 55, p. 102 615, 2020.

[106] Z. Wen, Y. Zheng, and Y. Li, "Analysis of decentralized energy transactions based on smart contract," in *2020 IEEE International Conference on Information Technology, Big Data and Artificial Intelligence (ICIBA)*, IEEE, vol. 1, 2020, pp. 819–824.

[107] W. Vickrey, "Counterspeculation, auctions, and competitive sealed tenders," *The Journal of Finance*, vol. 16, no. 1, pp. 8–37, 1961.

[108] D. Friedman, "The continuous double auction market institution: A survey," *Double Auction Markets: Theories, Institutions, and Experimental Evaluations, edited by D. Friedman, J. Geanakoplos, D. Lane, and J. Rust, Redwood City, CA: Addison-Wesley*, 1992.

[109] Y. Wang, W. Saad, Z. Han, H. V. Poor, and T. Başar, "A game-theoretic approach to energy trading in the smart grid," *IEEE Transactions on Smart Grid*, vol. 5, no. 3, pp. 1439–1450, 2014.

[110] M. Foti and M. Vavalis, "Blockchain based uniform price double auctions for energy markets," *Applied Energy*, vol. 254, p. 113 604, 2019.

[111] (Jun. 2022). The alberta electric system operator, [Online]. Available: https://www.aeso.ca/ (visited on 06/30/2022).

[112] Q. Yang, Y. Liu, Y. Cheng, Y. Kang, T. Chen, and H. Yu, "Federated learning," *Synthesis Lectures on Artificial Intelligence and Machine Learning*, vol. 13, no. 3, pp. 1–207, 2019.

[113] P. Ledger, "Power ledger white paper," *Power Ledger Pty Ltd, Australia, White paper*, vol. 8, 2017.

[114] R. Amer, W. Saad, H. ElSawy, M. M. Butt, and N. Marchetti, "Caching to the sky: Performance analysis of cache-assisted comp for cellular-connected uavs," in *2019 IEEE Wireless Communications and Networking Conference (WCNC)*, IEEE, 2019, pp. 1–6.

[115] E. Sortomme, M. M. Hindi, S. J. MacPherson, and S. Venkata, "Coordinated charging of plug-in hybrid electric vehicles to minimize distribution system losses," *IEEE Transactions on Smart Grid*, vol. 2, no. 1, pp. 198–205, 2010.

[116] H. Wang, J. Zhang, C. Lu, and C. Wu, "Privacy preserving in non-intrusive load monitoring: A differential privacy perspective," *IEEE Transactions on Smart Grid*, vol. 12, no. 3, pp. 2529–2543, 2020.

[117] A. Dorri, A. Hill, S. Kanhere, R. Jurdak, F. Luo, and Z. Y. Dong, "Peer-to-peer energytrade: A distributed private energy trading platform," in *2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, IEEE, 2019, pp. 61–64.

[118] K. Wüst and A. Gervais, "Do you need a blockchain?" In *2018 Crypto Valley Conference on Blockchain Technology (CVCBT)*, IEEE, 2018, pp. 45–54.

[119] J. Benet, "IPFS-content addressed, versioned, P2P file system (draft 3)," *arXiv preprint arXiv:1407.3561*, 2014.

[120] H. Huang, J. Lin, B. Zheng, Z. Zheng, and J. Bian, "When blockchain meets distributed file systems: An overview, challenges, and open issues," *IEEE Access*, vol. 8, pp. 50 574–50 586, 2020.

[121] M. Steichen, B. Fiz, R. Norvill, W. Shbair, and R. State, "Blockchain-based, decentralized access control for ipfs," in *2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, IEEE, 2018, pp. 1499–1506.

[122] M. Kuzlu, M. Pipattanasomporn, and S. Rahman, "Communication network requirements for major smart grid applications in HAN, NAN and WAN," *Computer Networks*, vol. 67, pp. 74–88, 2014.

[123] Z. Li, S. Bahramirad, A. Paaso, M. Yan, and M. Shahidehpour, "Blockchain for decentralized transactive energy management system in networked microgrids," *The Electricity Journal*, vol. 32, no. 4, pp. 58–72, 2019.

[124] R. Belchior, A. Vasconcelos, S. Guerreiro, and M. Correia, "A survey on blockchain interoperability: Past, present, and future trends," *ACM Comput. Surv.*, vol. 54, no. 8, Oct. 2021, ISSN: 0360-0300. DOI: 10.1145/3471140. [Online]. Available: https://doi.org/10.1145/3471140.

[125] Z. Liu, D. Wang, J. Wang, X. Wang, and H. Li, "A blockchain-enabled secure power trading mechanism for smart grid employing wireless networks," *IEEE Access*, vol. 8, pp. 177 745–177 756, 2020.

[126] I. Sestrem Ochôa, L. Augusto Silva, G. De Mello, N. M. Garcia, J. F. de Paz Santana, and V. R. Quietinho Leithardt, "A cost analysis of implementing a blockchain architecture in a smart grid scenario using sidechains," *Sensors*, vol. 20, no. 3, p. 843, 2020.

[127] X. Kong, J. Zhang, H. Wang, and J. Shu, "Framework of decentralized multi-chain data management for power systems," *CSEE Journal of Power and Energy Systems*, vol. 6, no. 2, pp. 458–468, 2019.

[128] H. Mehrjerdi, S. Lefebvre, M. Saad, and D. Asber, "A decentralized control of partitioned power networks for voltage regulation and prevention against disturbance propagation," *IEEE Transactions on Power Systems*, vol. 28, no. 2, pp. 1461–1469, 2012.

[129] H. Mehrjerdi, S. Lefebvre, M. Saad, and D. Asber, "Coordinated control strategy considering effect of neighborhood compensation for voltage improvement in transmission systems," *IEEE Transactions on Power Systems*, vol. 28, no. 4, pp. 4507–4515, 2013.

[130] J. Schiffer, T. Seel, J. Raisch, and T. Sezi, "Voltage stability and reactive power sharing in inverter-based microgrids with consensus-based distributed voltage control," *IEEE Transactions on Control Systems Technology*, vol. 24, no. 1, pp. 96–109, 2015.

[131] (Sep. 2022). U. of washington. power systems test case archive, [Online]. Available: https://uofi.app.box.com/s/frjqsg9vpe6dvv7ufodd (visited on 09/30/2022).

[132] T. Van Cutsem and C. Vournas, *Voltage stability analysis of electric power systems*, 1998.

[133] P. Kundur, J. Paserba, V. Ajjarapu, G. Andersson, A. Bose, C. Canizares, N. Hatziargyriou, D. Hill, A. Stankovic, C. Taylor, *et al.*, "Definition and classification of power system stability IEEE/CIGRE joint task force on stability terms and definitions," *IEEE Transactions on Power Systems*, vol. 19, no. 3, pp. 1387–1401, 2004.

[134] A. Chakrabarti and S. Halder, *Power System Analysis: Operation And Control 3Rd Ed.* PHI Learning Pvt. Ltd., 2010.

[135] V. Ajjarapu, *Computational Techniques for Voltage Stability Assessment and Control*. Springer, 2007.

[136] S. Li, J. Hou, A. Yang, and J. Li, "Dnn-based distributed voltage stability online monitoring method for large-scale power grids," *Frontiers in Energy Research*, vol. 9, p. 80, 2021.

[137] N. H. Abbasy and H. M. Ismail, "A unified approach for the optimal PMU location for power system state estimation," *IEEE Transactions on Power Systems*, vol. 24, no. 2, pp. 806–813, 2009.

[138] D. Carrion and J. W. Gonzalez, "Optimal PMU location in electrical power systems under N-1 contingency," in *2018 International Conference on Information Systems and Computer Science (INCISCOS)*, IEEE, 2018, pp. 165–170.

[139] I Power *et al.*, "Ieee standard for synchrophasor measurements for power systems–amendment 1: Modification of selected performance requirements," *IEEE Std C37. 118.1 a-2014 (Amendment to IEEE Std C37. 118.1-2011)*, vol. 2014, pp. 1–25, 2014.

[140] Y. Gong and N. Schulz, "Synchrophasor-based real-time voltage stability index," in *2006 IEEE PES Power Systems Conference and Exposition*, IEEE, 2006, pp. 1029–1036.

[141] C. Ren, Y. Xu, Y. Zhang, and R. Zhang, "A hybrid randomized learning system for temporal-adaptive voltage stability assessment of power systems," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 6, pp. 3672–3684, 2019.

[142] H Khoshkhoo and S. Shahrtash, "On-line dynamic voltage instability prediction based on decision tree supported by a wide-area measurement system," *IET Generation, Transmission & Distribution*, vol. 6, no. 11, pp. 1143–1152, 2012.

[143] K. D. Dharmapala, A. Rajapakse, K. Narendra, and Y. Zhang, "Machine learning based real-time monitoring of long-term voltage stability using voltage stability indices," *IEEE Access*, vol. 8, pp. 222 544–222 555, 2020.

[144] K. Sajan, V. Kumar, and B. Tyagi, "Genetic algorithm based support vector machine for on-line voltage stability monitoring," *International Journal of Electrical Power & Energy Systems*, vol. 73, pp. 200–208, 2015.

[145] J. De Kock and C. Strauss, *Practical power distribution for industry*. Elsevier, 2004.

[146] S. S. Biswas, "Synchrophasor based voltage stability monitoring and control of power systems," PhD thesis, Electrical and Computer Engineering Department, Washington State University, 2014.

[147] R. E. Wilson, "Pmus [phasor measurement unit]," *IEEE Potentials*, vol. 13, no. 2, pp. 26–28, 1994.

[148] J. C. Corbett, J. Dean, M. Epstein, A. Fikes, C. Frost, J. J. Furman, S. Ghemawat, A. Gubarev, C. Heiser, P. Hochschild, *et al.*, "Spanner: Google's globally distributed database," *ACM Transactions on Computer Systems (TOCS)*, vol. 31, no. 3, pp. 1–22, 2013.

[149] W. Tong, X. Dong, Y. Shen, and X. Jiang, "A hierarchical sharding protocol for multi-domain iot blockchains," in *ICC 2019-2019 IEEE International Conference on Communications (ICC)*, IEEE, 2019, pp. 1–6.

[150] E. Madill, B. Nguyen, C. K. Leung, and S. Rouhani, "Scalesfl: A sharding solution for blockchain-based federated learning," *arXiv preprint arXiv:2204.01202*, 2022.

[151] S. Yuan, B. Cao, M. Peng, and Y. Sun, "Chainsfl: Blockchain-driven federated learning from design to realization," in *2021 IEEE Wireless Communications and Networking Conference (WCNC)*, IEEE, 2021, pp. 1–6.

[152] Y. Lu, X. Huang, K. Zhang, S. Maharjan, and Y. Zhang, "Low-latency federated learning and blockchain for edge association in digital twin empowered 6G networks," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 7, pp. 5098–5107, 2020.

[153] E. Kokoris-Kogias, P. Jovanovic, L. Gasser, N. Gailly, E. Syta, and B. Ford, "Omniledger: A secure, scale-out, decentralized ledger via sharding," in *2018 IEEE Symposium on Security and Privacy (SP)*, IEEE, 2018, pp. 583–598.

[154] L. Luu, V. Narayanan, C. Zheng, K. Baweja, S. Gilbert, and P. Saxena, "A secure sharding protocol for open blockchains," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, 2016, pp. 17–30.

[155] G. Danezis and S. Meiklejohn, "Centrally banked cryptocurrencies," *arXiv preprint arXiv:1505.06895*, 2015.

[156] J. Xie, F. R. Yu, T. Huang, R. Xie, J. Liu, and Y. Liu, "A survey on the scalability of blockchain systems," *IEEE Network*, vol. 33, no. 5, pp. 166–173, 2019.

[157] R. D. Zimmerman, C. E. Murillo-Sánchez, and R. J. Thomas, "Matpower: Steady-state operations, planning, and analysis tools for power systems research and education," *IEEE Transactions on power systems*, vol. 26, no. 1, pp. 12–19, 2010.

[158] W. F. Tinney and C. E. Hart, "Power flow solution by newton's method," *IEEE Transactions on Power Apparatus and Systems*, no. 11, pp. 1449–1460, 1967.

[159] B. Sereeter, C. Vuik, and C. Witteveen, "On a comparison of Newton–Raphson solvers for power flow problems," *Journal of Computational and Applied Mathematics*, vol. 360, pp. 157–169, 2019.

[160] B. Stott and O. Alsac, "Fast decoupled load flow," *IEEE Transactions on Power Apparatus and Systems*, no. 3, pp. 859–869, 1974.

[161] R. A. Van Amerongen, "A general-purpose version of the fast decoupled load flow," *IEEE Transactions on Power Systems*, vol. 4, no. 2, pp. 760–770, 1989.

[162] A. Glimn and G. Stagg, "Automatic calculation of load flows," *Transactions of the American Institute of Electrical Engineers. Part III: Power Apparatus and Systems*, vol. 76, no. 3, pp. 817–825, 1957.

[163] O. Alsac and B. Stott, "Optimal load flow with steady-state security," *IEEE Transactions on Power Apparatus and Systems*, no. 3, pp. 745–751, 1974.

Appendix A: Voltage Stability Index Threshold Estimation

To estimate a critical point and create data for testing the LocalController algorithm, we need to estimate the threshold for the Voltage Stability Index. Therefore, we defined multiple scenarios by increasing the load of different buses. We used the monitoring and controlling algorithm described in Chapter 4 to calculate VSI and specify the required reactive power needed to be injected at each bus. These scenarios helped us to approximate the VSI threshold.

In this chapter, we first talk about the base model that we consider as a stable grid. Then we explain three different scenarios and actions we take as a Volt-Var Control to estimate the thresholds. For all the calculations and simulation, we used the smart contracts described in Chapter 4.

A.1 Base Model

As we described in Chapter 4, we used the IEEE 30 bus transmission system data for evaluating our model. We consider the original value of the IEEE 30 system as a stable grid and our base model. Then we split the grid into three different groups and calculated the VSI for each group and their combination to explore the VSI value. Figure A.1 shows the information of each 30 buses. The buses can have one of these three types: Load bus (PQ), generator bus(PV), and reference bus. VSI must be calculated for the load buses, and the value of VSI is 1 for the buses that do not have any reactive or active load. We specified the group in the figure with three different

colours and calculated the minimum VSI value for each group. When we merge two groups with each other, the minimum VSI value changes a little bit. This is because we supposed the buses, separating groups from each other, are tie buses. Based on the flow of the power in tie buses, we considered them as a virtual load or generator. But when we merged groups, we considered them as a load bus. This fact led to the difference in VSI values.

It also needs to be mentioned that the VSI value is very sensitive to small changes in the system, and if we suddenly put a large amount of the load on the grid, it may not give us a very good estimation about a critical point.

A.2 Scenario 1

In the first scenario, we increased the load at bus 3 by multiplying it by 22. The active power (P load) increased from 2.4 to 52.8 kW, and the reactive power (Q load) increased from 1.2 to 26.4 kVAr at bus 3. The voltage magnitude dropped from 0.98 to 0.95, which led to decreasing the VSI value. Bus 3 is located in group 1, and when the voltage decrease at the bus, it just affects the minimum VSI value for group 1, and the minimum VSI for the other groups does not change.

As a control action, we consider two different scenarios:

First, we supposed the VVC at bus 3 had enough reactive power. In this case, injecting the 11.072 kVAr reactive power at bus 3 will restore the voltage and increase the VSI from 0.95 to 0.96. Figure A.2 shows the calculated VSI values in this scenario.

Second, we supposed there is no VVC available on bus 3, and we need to inject power from bus 4. In this case, we need to inject reactive power two times from VVC located at bus 4 to increase the voltage. Figure A.3 shows the calculated voltage and VSI for this scenario.

Based on these two scenarios and several other samples in group 1, we decided that the best VSI threshold for group 1 is 0.9015.

Original case30	1= PQ , 2=PV, 3=reference				red=group1 , blue=group2 green=group3				
Bus	**Type**	**P load**	**Q load**	**Voltage magnitude**	**Voltage angle**	**VSI**	**VSI -merged group1_2**	**VSI-merged group1_3**	**VSI-merged group2_3**
1	3	0	0	1	0	-	-	-	
2	2	21.7	12.7	1	-0.4154907169	-	-	-	-
3	1	2.4	1.2	0.9831382892	-1.522073935	0.9982175118	0.9981906399	0.9982004369	0.9982175118
4	1	7.6	1.6	0.9800929954	-1.794727651	0.9979435511	0.9981906399	0.9978653502	0.9979435511
5	1	0	0	0.9824061968	-1.863822669	1	1	1	1
6	1	0	0	0.9731840212	-2.266956702	1	1	1	1
7	1	22.8	10.9	0.9673554478	-2.651836762	0.8697239667	0.9308645672	0.8697239667	0.8860449301
8	1	30	30	0.9606237084	-2.725769431	0.7690864669	0.8856441239	0.7690864669	0.7957263093
9	1	0	0	0.9805061171	-2.99693309	1	1	1	1
10	1	5.8	2	0.9844042958	-3.37493594	0.9990497389	0.9990497389	0.9982137498	0.9981460586
11	1	0	0	0.9805061171	-2.99693309	1	1	1	1
12	1	11.2	7.5	0.9854683171	-1.536911577	0.9948521908	0.9948521908	0.9929903557	0.992968575
13	2	0	0	1	1.476163259	-	-	-	-
14	1	6.2	1.6	0.9766768341	-2.308035423	0.9519880715	0.9519880715	0.9629687894	0.9695416595
15	1	8.2	2.5	0.980229029	-2.311835386	0.9976196613	0.9976196613	0.9971365176	0.9963809734
16	1	3.5	1.8	0.9773956552	-2.644486203	0.9985099638	0.9985099638	0.9980652873	0.9980432088
17	1	9	5.8	0.9768654099	-3.392339211	0.9781554026	0.9781554026	0.9775716799	0.9812206816
18	1	3.2	0.9	0.9684403291	-3.478387713	0.9956795035	0.9956795035	0.9954382786	0.995485975
19	1	9.5	3.4	0.9652870397	-3.958204685	0.9863339799	0.9863339799	0.9850957556	0.9854186738
20	1	2.2	0.7	0.9691663508	-3.871024321	0.9904464767	0.9904464767	0.9885306369	0.9883185209
21	1	17.5	11.2	0.9933832968	-3.488393316	0.8478958439	0.8478958439	0.9091415721	0.9880255047
22	2	0	0	1	-3.392729009	-	-	-	-
23	2	3.2	1.6	1	-1.589227916	-	-	-	-
24	1	8.7	6.7	0.9885662962	-2.631461464	0.975977914	0.975977914	0.975977914	0.975977914
25	1	0	0	0.9902148369	-1.689988884	1	1	1	1
26	1	3.5	2.3	0.9721941498	-2.139345979	0.9283226322	0.9283226322	0.9283226322	0.9283226322
27	2	0	0	1	-0.8284393157	-	-	-	-
28	1	0	0	0.9747148998	-2.265928622	1	1	1	1
29	1	2.4	0.9	0.9795967047	-2.128498172	0.9950062022	0.9950062022	0.9950062022	0.9950062022
30	1	10.6	1.9	0.9678828792	-3.041523574	0.9039539632	0.9039539632	0.9039539632	0.9039539632
					group1 min:	0.9979435511	0.9981906399	0.9978653502	0.9979435511
					group2 min:	0.7690864669	0.8856441239	0.7690864669	0.7957263093
					group3 min:	0.8478958439	0.8478958439	0.9091415721	0.9695416595

Figure A.1: Voltage Stability Index Value (VSI) for the base model (original IEEE 30 bus system value).

increse bus3 load *22		1= PQ 2=PV 3=reference			red=group1 , blue=group2 green=group3							
Bus	**Type**	**P load**	**Q load**	**Voltage mag**	**Voltage angl**	**VSI**				**Voltage mag**	**Voltage angl**	**VSI**
1	3	0	0	1	0	-				1	0	-
2	2	21.7	12.7	1	-1.441502525	-				1	-1.432109981	-
3	1	52.8	26.4	0.9556034763	-4.195927899	0.8788443301				0.9631684222	-4.289535685	0.9018516321
4	1	7.6	1.6	0.9632409726	-3.959646382	0.9909276513				0.9676948771	-4.014296003	0.9936725806
5	1	0	0	0.9770752239	-3.347026237	1				0.9785278048	-3.351547481	1
6	1	0	0	0.9617891911	-4.200435515	1				0.9646838065	-4.231866724	1
7	1	22.8	10.9	0.9581800772	-4.411590053	0.8672170183				0.9605226201	-4.430672237	0.8678638964
8	1	30	30	0.9492613467	-4.674470014	0.7635254575				0.9521474485	-4.702159924	0.7649568644
9	1	0	0	0.9748822933	-4.987353771	1				0.9763271798	-5.017401139	1
10	1	5.8	2	0.9818124803	-5.391133304	0.9991207779				0.9824980041	-5.421401652	0.9991173791
11	1	0	0	0.9748822933	-4.987353771	1				0.9763271798	-5.017401139	1
12	1	11.2	7.5	0.9809836968	-3.589171275	0.994248183				0.9821131812	-3.64087348	0.9942194954
13	2	0	0	1	-0.5623092053	-				1	-0.6174957051	-
14	1	6.2	1.6	0.9729636801	-4.368025203	0.9516209132	activate VVC at bus3 ==> inject 11.0712522			0.9738906829	-4.413682728	0.9517129691
15	1	8.2	2.5	0.9772915374	-4.392182198	0.9974948369				0.9780429084	-4.428553821	0.9974879022
16	1	3.5	1.8	0.9737110303	-4.685453381	0.9984059933				0.9746624462	-4.726954713	0.9983949963
17	1	9	5.8	0.973942294	-5.417814201	0.9764807594				0.9747119303	-5.450823991	0.9763372631
18	1	3.2	0.9	0.9656070676	-5.540913333	0.9954906284				0.9663468543	-5.573881663	0.9954793884
19	1	9.5	3.4	0.9625114822	-6.008758104	0.9857246934				0.9632405052	-6.040067822	0.9856880361
20	1	2.2	0.7	0.9664330818	-5.913177545	0.9910811162				0.967151851	-5.944139208	0.9910954544
21	1	17.5	11.2	0.9928057965	-5.55711917	0.8452671615				0.9929587149	-5.574121187	0.8454235362
22	2	0	0	1	-5.47839708	-				1	-5.491106499	-
23	2	3.2	1.6	1	-3.719957749	-				1	-3.735162908	-
24	1	8.7	6.7	0.9885470377	-4.71832063	0.975977914				0.9885462617	-4.733765427	0.9764068473
25	1	0	0	0.9902101624	-3.722746356	1				0.9902096232	-3.743354415	1
26	1	3.5	2.3	0.9721893867	-4.172107774	0.9283219299				0.9721888373	-4.192716332	0.9283218488
27	2	0	0	1	-2.827894853	-				1	-2.851680125	-
28	1	0	0	0.9644792152	-4.228908469	1				0.967074916	-4.253811339	1
29	1	2.4	0.9	0.9795967047	-4.127953709	0.9950062022				0.9795967047	-4.151738981	0.9950062022
30	1	10.6	1.9	0.9678828792	-5.040979111	0.9039539632				0.9678828792	-5.064764383	0.9039539632
					group1 min:	0.8788443301					group1 min:	0.9018516321
					group2 min:	0.7635254575					group2 min:	0.7649568644
					group3 min:	0.8452671615					group3 min:	0.8454235362

Figure A.2: Voltage Stability Index (VSI) and Voltage Magnitude value for scenario 1. We supposed there is enough reactive power available at VVC at bus 3.

increse bus3 load *22		1= PQ 2=PV 3=reference			red=group1 , blue=group2 green=group3																		
Bus	**Type**	**P load**	**Q load**	**Voltage mag**	**Voltage angl**	**VSI**						**Voltage mag**	**Voltage angl**	**VSI**					**Voltage mag**	**Voltage angl**	**VSI**
1	3	0	0	1	0	-						1	0	-					1	0	-
2	2	21.7	12.7	1	-1.441502525	-						1	-1.412521175	-					1	-1.410364191	-
3	1	52.8	26.4	0.9556034763	-4.195927899	0.8788443301						0.9772701497	-4.479134185	0.8841569674					0.9796826563	-4.511990646	0.989338279
4	1	7.6	1.6	0.9632409726	-3.959646382	0.9909276513						0.9891644759	-4.330355262	0.9990011286					0.9920528574	-4.372881316	0.9990959095
5	1	0	0	0.9770752239	-3.347026237	1						0.9855261566	-3.411477299	1					0.9864668911	-3.420164264	1
6	1	0	0	0.9617891911	-4.200435515	1						0.9786191451	-4.432434334	1					0.9804930115	-4.459865182	1
7	1	22.8	10.9	0.9581800772	-4.411590053	0.8672170183						0.9717970505	-4.567838006	0.8709120997					0.9733126498	-4.58687118	0.8713138066
8	1	30	30	0.9492613467	-4.674470014	0.7635254575						0.9660381129	-4.885170593	0.7716676396					0.9679055227	-4.910294662	0.7725478469
9	1	0	0	0.9748822933	-4.987353771	1						0.9832846302	-5.212945044	1					0.9842201904	-5.23975834	1
10	1	5.8	2	0.9818124803	-5.391133304	0.9991207779						0.9858007493	-5.61879642	0.9951986471					0.9862448757	-5.645882167	0.99910075
11	1	0	0	0.9748822933	-4.987353771	1						0.9832846302	-5.212945044	1					0.9842201904	-5.23975834	1
12	1	11.2	7.5	0.9809836968	-3.589171275	0.994248183						0.9875549181	-3.942436002	0.9951986471					0.988287254	-3.983316201	0.9952006336
13	2	0	0	1	-0.5623092053	-						1	-0.9357333378	-					1	-0.9788435941	-
14	1	6.2	1.6	0.9729636801	-4.368025203	0.9516209132	activate VVC at bus4 ==> inject 55.35002089 5					0.9783557392	-4.686283127	0.9521527117	activate VVC at bus4 ==> inject 6.357681854588881				0.9789565849	-4.723330697	0.9522114273
15	1	8.2	2.5	0.9772915374	-4.392182198	0.9974948369						0.9816618813	-4.656545063	0.9974534357					0.9821486582	-4.687651188	0.9974491091
16	1	3.5	1.8	0.9737110303	-4.685453381	0.9984059933						0.9792460231	-4.979084711	0.9917410563					0.9798625894	-5.013420652	0.9917086577
17	1	9	5.8	0.973942294	-5.417814201	0.9764807594						0.9784198181	-5.661640135	0.9756004931					0.9789184454	-5.690502683	0.9755022295
18	1	3.2	0.9	0.9656070676	-5.540913333	0.9954906284						0.9699104981	-5.785097479	0.9954211656					0.970389733	-5.81398592	0.9954134901
19	1	9.5	3.4	0.9625114822	-6.008758104	0.9857246934						0.9667523986	-6.243056137	0.9854981562					0.967224644	-6.270859452	0.9854731068
20	1	2.2	0.7	0.9664330818	-5.913177545	0.9910811162						0.9706144131	-6.14530743	0.9911742327					0.9710800255	-6.172875153	0.9911842024
21	1	17.5	11.2	0.9928057965	-5.55711917	0.8452671615						0.9936952456	-5.707966479	0.8458909722					0.9937942609	-5.726545617	0.8459237964
22	2	0	0	1	-5.47839708	-						1	-5.604302022	-					1	-5.620106284	-
23	2	3.2	1.6	1	-3.719957749	-						1	-3.861169643	-					1	-3.878643606	-
24	1	8.7	6.7	0.9885470377	-4.71832063	0.975977914						0.9885421225	-4.860199523	0.9761890582					0.9885415783	-4.877759625	0.9761596446
25	1	0	0	0.9902101624	-3.722746356	1						0.9902068431	-3.893838653	1					0.9902064661	-3.914626252	1
26	1	3.5	2.3	0.9721893867	-4.172107774	0.9283219299						0.9721860045	-4.343203141	0.9283214311					0.9721856204	-4.363991089	0.9283213745
27	2	0	0	1	-2.827894853	-						1	-3.016963299	-					1	-3.039736886	-
28	1	0	0	0.9644792152	-4.228908469	1						0.9795698439	-4.423626915	1					0.9812499013	-4.44699487	1
29	1	2.4	0.9	0.9795967047	-4.127953709	0.9950062022						0.9795967047	-4.317022156	0.9950062022					0.9795967047	-4.339795742	0.9950062022
30	1	10.6	1.9	0.9678828792	-5.040979111	0.9039539632						0.9678828792	-5.230047557	0.9039539632					0.9678828792	-5.252821144	0.9039539632
					group1 min:	0.8788443301							group1 min:	0.8841569674						group1 min:	0.989338279
					group2 min:	0.7635254575							group2 min:	0.7716676396						group2 min:	0.7725478469
					group3 min:	0.8452671615							group3 min:	0.8458909722						group3 min:	0.8459237964

Figure A.3: Voltage Stability Index (VSI) and Voltage Magnitude value for scenario 1. We supposed there is not enough reactive power available at VVC at bus3, and we need to inject power from VVC at bus 4. It takes two steps of injection to increase the VSI value and voltage.

A.3 Scenario 2

In the second scenario, we increased the load at bus 14 by multiplying it by 4. The active load (P load) increased from 6.2 to 24.8 kW, and the reactive load (Q load) increased from 1.6 to 6.4 kVAr at bus 14. The voltage magnitude dropped from 0.97 to 0.94, which led to decreasing the VSI value. Bus 14 is located in group 3, and when the voltage decrease at the bus, the minimum VSI value for the group 3 drops from 0.847 to 0.79, and the minimum VSI for the other groups does not change.

As a control action, we supposed the VVC at bus 14 had not enough reactive power, and we used VVC at bus 15 to restore the voltage. In this case, we need to inject reactive power at bus 14 three times to correct the voltage and increase the VSI from 0.79 to 0.801. Figure A.4 and Figure A.5 show the calculated VSI values and voltages in this scenario.

Based on this scenario and several other samples in group 3, we decided that the best VSI threshold for group 3 is 0.80.

A.4 Scenario 3

In the third scenario, we increased the load at bus 30 by multiplying it by 2. The active load (P load) increased from 10.6 to 17.4 kW, and the reactive load (Q load) increased from 1.9 to 13.4 kVAr at bus 30. The voltage magnitude dropped from 0.96 to 0.90, which led to decreasing the VSI value. Bus 30 is located in group 2, and when the voltage decrease at the bus, the minimum VSI value for the group 2 decrease from 0.76 to 0.69, and the minimum VSI for the other groups does not change very much. There is a little drop in the minimum VSI value of group 3, which is above the threshold we defined in scenario 2.

As a control action, we supposed the VVC at bus 30 had enough reactive power. In this case, we need to inject reactive power one time at bus 30 to correct the voltage and increase the VSI from 0.69 to 0.76. Figure A.6 shows the calculated VSI values

increse bus14 load *4		1= PQ 2=PV 3=reference			red=group1 , blue=group2 green=group3								
Bus	**Type**	**P load**	**Q load**	**Voltage magnitude**	**Voltage angle**	**VSI**					**Voltage magnitude**	**Voltage angle**	**VSI**
1	3	0	0	1	0	-					1	0	-
2	2	21.7	12.7	1	-0.89531227	-					1	-0.892776006	-
3	1	2.4	1.2	0.98015719	-2.328142717	0.9987580721					0.9806391005	-2.334053388	0.9987351935
4	1	7.6	1.6	0.9766439674	-2.7751042	0.9984849001					0.9772277421	-2.781964104	0.9985056229
5	1	0	0	0.9811161308	-2.610614368	1					0.9813508813	-2.60895086	1
6	1	0	0	0.9701707923	-3.290711602	1					0.970640382	-3.291962195	1
7	1	22.8	10.9	0.964948263	-3.566632434	0.8690731761					0.9653282346	-3.566458471	0.8691762264
8	1	30	30	0.9575890725	-3.761439803	0.7676206005					0.9580581337	-3.76184731	0.7678480888
9	1	0	0	0.9791868979	-4.576946595	1					0.9798134647	-4.56669612	1
10	1	5.8	2	0.984103783	-5.241108951	0.9995214643					0.9848091967	-5.224767369	0.999497963
11	1	0	0	0.9791868979	-4.576946595	1					0.9798134647	-4.56669612	1
12	1	11.2	7.5	0.9802447817	-4.181851438	0.9986149556	activate VVC at bus15 inject =7.341455832851773				0.9831979117	-4.202012244	0.9986160952
13	2	0	0	1	-1.152705574	-					1	-1.181973159	-
14	1	24.8	6.4	0.9472775314	-6.247758289	0.7958466821					0.9523060381	-6.28732997	0.7979969907
15	1	8.2	2.5	0.9733137154	-5.258680562	0.9934535578					0.9797615305	-5.39038581	0.995506248
16	1	3.5	1.8	0.9744844644	-4.961896404	0.9966422433					0.9764930296	-4.963477237	0.9844308819
17	1	9	5.8	0.9758723389	-5.391888802	0.9516199417					0.9769682934	-5.379550211	0.9527350914
18	1	3.2	0.9	0.9641145087	-6.054107693	0.9898447446					0.9686093722	-6.123092542	0.9917449294
19	1	9.5	3.4	0.9622769821	-6.308488445	0.9671084121					0.9655926907	-6.345821329	0.8949784929
20	1	2.2	0.7	0.9668091847	-6.098223032	0.996238372					0.9694714419	-6.122154919	0.9961884599
21	1	17.5	11.2	0.9933257719	-5.402303035	0.8144376529					0.9934830309	-5.37195642	0.8148167601
22	2	0	0	1	-5.320632117	-					1	-5.285755868	-
23	2	3.2	1.6	1	-4.294481459	-					1	-4.252045151	-
24	1	8.7	6.7	0.9882223089	-4.75230322	0.9791767929					0.988226804	-4.718465141	0.9790809223
25	1	0	0	0.9900893336	-3.404578582	1					0.9900913132	-3.383599526	1
26	1	3.5	2.3	0.9720662679	-3.854051763	0.9283037717					0.972068285	-3.833070876	0.9283040692
27	2	0	0	1	-2.292957534	-					1	-2.279890571	-
28	1	0	0	0.9717956466	-3.344860378	1					0.9722233761	-3.343583543	1
29	1	2.4	0.9	0.9795967047	-3.59301639	0.9950062022					0.9795967047	-3.579949427	0.9950062022
30	1	10.6	1.9	0.9678828792	-4.506041792	0.9039539632					0.9678828792	-4.492974829	0.9039539632
					group1 min:	0.9984849001						group1 min:	0.9985056229
					group2 min:	0.7676206005						group2 min:	0.7678480888
					group3 min:	0.7958466821						group3 min:	0.7979969907

Figure A.4: Voltage Stability Index (VSI) and Voltage Magnitude value for scenario 2 - part 1 . We supposed there is not enough reactive power available at VVC at bus 14, and we need to inject power from VVC at bus 15. It takes three steps of injection to increase the VSI value and voltage.

				Voltage magnitude	Voltage angle	VSI					Voltage magnitude	Voltage angle	VSI
				1	0	-					1	0	-
				1	-0.8914494695	-					1	-0.8909014483	-
				0.9810400465	-2.340364251	0.9987351935					0.9813750286	-2.34663303	0.9987272015
				0.9777137519	-2.789362324	0.9985056229					0.9781200283	-2.796752056	0.9985146151
				0.9815463165	-2.608817024	1					0.9817096898	-2.609598701	1
				0.9710306295	-3.294755702	1					0.9713563592	-3.298341863	1
				0.9656440423	-3.567866589	0.8692617826					0.9659076646	-3.570152512	0.8693331368
				0.958447881	-3.763955141	0.7680368568					0.9587731501	-3.76697957	0.7681942197
				0.9803384443	-4.560856868	1					0.9807796913	-4.557896474	1
				0.9854022379	-5.214341317	0.9994782986					0.9859021038	-5.207890739	0.9994619526
				0.9803384443	-4.560856868	1					0.9807796913	-4.557896474	1
activate VVC at bus15 inject =6.243750153281506				0.9856783484	-4.222976951	0.9986196276	activate VVC at bus15 inject =5.313680241933815				0.9877674951	-4.243480023	0.9986239344
				1	-1.210544744	-					1	-1.237425024	-
				0.956527201	-6.325141761	0.7997759389					0.9600807705	-6.360211959	0.8012553839
				0.9851735846	-5.505106673	0.9958132403					0.989729339	-5.604619936	0.9963666275
				0.9781804188	-4.968565398	0.9847109692					0.9796018457	-4.975506678	0.9849338229
				0.9778893568	-5.372630678	0.9536125891					0.9786654934	-5.369235514	0.9543116677
				0.9723813475	-6.185249693	0.9932077183					0.9755558947	-6.240574487	0.9942247118
				0.9683750342	-6.381243749	0.8955811233					0.9707165996	-6.413947682	0.8960842747
				0.9717055285	-6.146169501	0.9961507195					0.9735857429	-6.169158939	0.9961220177
				0.9936152489	-5.349880582	0.8150540926					0.993726701	-5.333697233	0.8151966714
				1	-5.259903424	-					1	-5.240558763	-
				1	-4.221606979	-					1	-4.19963493	-
				0.9882297181	-4.693962678	0.9790056809					0.988231573	-4.676088701	0.9789460816
				0.9900926066	-3.36909855	1					0.9900934419	-3.35908603	1
				0.9720696029	-3.818568703	0.9283042636					0.9720704541	-3.808555411	0.9283043892
				1	-2.271543408	-					1	-2.26636791	-
				0.9725784465	-3.344363095	1					0.9728745392	-3.346329452	1
				0.9795967047	-3.571602265	0.9950062022					0.9795967047	-3.566426766	0.9950062022
				0.9678828792	-4.484627666	0.9039539632					0.9678828792	-4.479452168	0.9039539632
					group1 min:	0.9985056229						group1 min:	0.9985146151
					group2 min:	0.7680368568						group2 min:	0.7681942197
					group3 min:	0.7997759389						group3 min:	0.8012553839

Figure A.5: Voltage Stability Index (VSI) and Voltage Magnitude value for scenario 2 - part 2 . We supposed there is not enough reactive power available at VVC at bus 14, and we need to inject power from VVC at bus 15. It takes three steps of injection to increase the VSI value and voltage.

in this scenario.

Based on this scenario and several other samples in group 2, we decided that the best VSI threshold for group 2 is 0.76.

increse bus30 load *2		1= PQ 2=PV 3=reference			red=group1 , blue=group2 green=group3								
Bus	**Type**	**P load**	**Q load**	**Voltage magnitude**	**Voltage angle**	**VSI**					**Voltage magnitude**	**Voltage angle**	**VSI**
1	3	0	0	1	0	-					1	0	-
2	2	21.7	12.7	1	-0.6131962462	-					1	-0.6022903537	-
3	1	2.4	1.2	0.9824590396	-1.830799787	0.9984243039					0.982498001	-1.813798075	0.9984132418
4	1	7.6	1.6	0.9793250677	-2.169840461	0.9981349887					0.9793690444	-2.149179711	0.9981247391
5	1	0	0	0.982031284	-2.187700261	1					0.9820531606	-2.169853648	1
6	1	0	0	0.9722252019	-2.72353159	1					0.9722801929	-2.698388368	1
7	1	22.8	10.9	0.96659605	-3.055955471	0.8695191859					0.9666397289	-3.03369612	0.8695309775
8	1	30	30	0.9595630357	-3.20990185	0.7685756948					0.9596239034	-3.183243087	0.7686050517
9	1	0	0	0.9801939592	-3.55574327	1					0.9802122669	-3.524936509	1
10	1	5.8	2	0.9844495766	-3.986239558	0.9991207779					0.9844475699	-3.952528302	0.9991079564
11	1	0	0	0.9801939592	-3.55574327	1					0.9802122669	-3.524936509	1
12	1	11.2	7.5	0.9856182696	-2.079270039	0.9956288104	activate VVC at bus30 inject =8.220514057978963				0.9856105425	-2.049367816	0.9955904513
13	2	0	0	1	0.9333459632	-					1	0.9632718257	-
14	1	6.2	1.6	0.9768912514	-2.882858304	0.9520091454					0.9768799152	-2.851162061	0.9520080316
15	1	8.2	2.5	0.9802819433	-2.919800096	0.9978151326					0.9802794603	-2.886271649	0.9978049629
16	1	3.5	1.8	0.9774800977	-3.216171536	0.9985867982					0.977475976	-3.184648771	0.9985827094
17	1	9	5.8	0.9769150126	-3.992049289	0.9793298458					0.9769127866	-3.958978196	0.9792672936
18	1	3.2	0.9	0.9684901765	-4.087445628	0.9956883149					0.9684878881	-4.053856666	0.9956878291
19	1	9.5	3.4	0.9653356227	-4.567935744	0.986362416					0.9653334176	-4.534309672	0.9863608478
20	1	2.2	0.7	0.9692142031	-4.481148803	0.990416768					0.9692120428	-4.447501364	0.9904183992
21	1	17.5	11.2	0.9934036954	-4.144270082	0.812198773					0.9934026831	-4.108085596	0.8138288032
22	2	0	0	1	-4.061448676	-					1	-4.024551189	-
23	2	3.2	1.6	1	-2.319709596	-					1	-2.279410512	-
24	1	8.7	6.7	0.9885300571	-3.524015095	0.9695479592					0.9885336087	-3.47482595	0.9699537618
25	1	0	0	0.9901325531	-3.187542977	1					0.9901394551	-3.105144289	1
26	1	3.5	2.3	0.9721103066	-3.636976177	0.9283102675					0.9721173394	-3.554571105	0.9283113048
27	2	0	0	1	-2.698259188	-					1	-2.595429802	-
28	1	0	0	0.9731265151	-2.882135478	1					0.9732173959	-2.848226753	1
29	1	2.4	0.9	0.9498265863	-4.121255057	0.9987061525					0.96590447	-4.430902645	0.9979501839
30	1	17.4	13.4	0.9064834884	-5.20946768	0.6943676801					0.9400210951	-5.969374354	0.8124576973
					group1 min:	0.9981349887						group1 min:	0.9981247391
					group2 min:	0.6943676801						group2 min:	0.7686050517
					group3 min:	0.812198773						group3 min:	0.8138288032

Figure A.6: Voltage Stability Index (VSI) and Voltage Magnitude value for scenario 3. We supposed there is enough reactive power available at VVC at bus 30.