electronics

MDPI

Article

Smart Microgrid Energy Market: Evaluating Distributed Ledger Technologies for Remote and Constrained Microgrid Deployments †

Lehlogonolo P. I. Ledwaba [1,*], Gerhard P. Hancke [1], Sherrin J. Isaac [2] and Hein S. Venter [3]

[1] Department of Computer Science, City University of Hong Kong, Kowloon, Hong Kong 999077, China; gp.hancke@cityu.edu.hk
[2] NextGen Enterprises and Institutions, Council for Scientific and Industrial Research, Pretoria 0001, South Africa; SIsaac@csir.co.za
[3] Department of Computer Science, University of Pretoria, Pretoria 0028, South Africa; hventer@cs.up.ac.za
[*] Correspondence: lpledwaba2-c@my.cityu.edu.hk

† This article is an expansion upon the following conference publication: Ledwaba, L.P.I.; Hancke, G.P.; Isaac, S.J.; Venter, H.S. Developing a Secure, Smart Microgrid Energy Market using Distributed Ledger Technologies. In Proceedings of the 2019 IEEE 17th International Conference on Industrial Informatics (INDIN), Helsinki, Finland, 22–25 July 2019; pp. 1725–1728, doi:10.1109/INDIN41052.2019.8972018.

Abstract: The increasing strain on ageing generation infrastructure has seen more frequent instances of scheduled and unscheduled blackouts, rising reliability on fossil fuel based energy alternatives and a slow down in efforts towards achieving universal access to electrical energy in South Africa. To try and relieve the burden on the National Grid and still progress electrification activities, the smart microgrid model and secure energy trade paradigm is considered—enabled by the Industrial IoT (IIoT) and distributed ledger technologies (DLTs). Given the high availability requirements of microgrid operations, the limited resources available on IIoT devices and the high processing and energy requirements of DLT operations, this work aims to determine the effect of native DLT algorithms when implemented on IIoT edge devices to assess the suitability of DLTs as a mechanism to establish a secure, energy trading market for the Internet of Energy. Metrics such as the node transaction time, operating temperature, power consumption, processor and memory usage are considered towards determining possible interference on the edge node operation. In addition, the cost and time required for mining operations associated with the DLT-enabled node are determined in an effort to predict the cost to end users—in terms of fees payable and mobile data costs—as well as predicting the microgrid's growth and potential blockchain network slowdown.

Keywords: blockchain; distributed ledger technology; Industry 4.0; Industrial Internet of Things; performance testing; Raspberry Pi; smart microgrid; smart contracts; security

check for updates

Citation: Ledwaba, L.P.I.; Hancke, G.P.; Isaac, S.J.; Venter, H.S. Smart Microgrid Energy Market: Evaluating Distributed Ledger Technologies for Remote and Constrained Microgrid Deployments. *Electronics* **2021**, *10*, 714. https://doi.org/10.3390/electronics10060714

Academic Editors: Sang-Soo Yeo, Damien Sauveron and Rameez Asif

Received: 3 February 2021
Accepted: 16 March 2021
Published: 18 March 2021

1. Motivation and Incitement

Governments in emerging economies are striving towards meeting the goals of universal access to electrical energy by 2030. One of the main aims is to ensure electrical energy is available to rural communities. As more households connect to the national grid, however, the increased load results in instability and supply deficits to the energy supply chain. This is often the result of insufficient and ageing generation and distribution infrastructure [1].

In response to the ongoing energy crisis, a collaborative effort by the Council of Scientific and Industrial Research (CSIR) are investigating a smart microgrid solution enabled by a secure, energy trade marketplace for rural and informal settlements in South Africa. Allowing households to have renewable, self-generation facilities would improve the reliability of the electrical energy supply, facilitate better economic growth and reduce the economic inequality between its citizens. This work forms part of the exploratory work needed as a foundation towards developing a secure energy marketplace for the greater smart microgrid solution. This work aims to conduct a feasibility study in order to assess the capabilities of native DLT algorithms when run in the IIoT context. The results of this work shall be used as part of the design and decision making process, establishing if DLTs are worthwhile solutions to use as part of the microgrid energy market. This work looks to running native DLT code implementations on a popular IIoT edge device platform in order to generate an initial set of operational and performance metrics, which can be considered further in the microgrid design.

South Africa has an area of 1.22 million km^2 [2] and the existing power grid has transmission lines that cover 31,107 km of the country while distribution lines cover 48,278 km of the country (2015 figures). The country relies on 15 coal power stations, 1 nuclear, 4 gas turbines, 6 hydroelectric, 3 pump storage, 2 wind and 1 solar plant for a population of 58.78 million people. Of the 32 generation facilities, 20 rely on fossil fuels (62.5% of energy generation). Coal power stations make up 46.88% of the generation capacity, leading to a high greenhouse emissions and a high carbon footprint for the country [3].

The results given in the last census conducted by Statistics South Africa are given in Figure 1. An increase in household electrification was observed across the nine provinces in the country over an eight year period; however, in the same period, the general user satisfaction in the quality of electrical energy supplied was found to have decreased [4,5]. The observed decrease in satisfaction could be attributed to South Africa's grid instability. Grid instability has resulted in increased blackouts—both scheduled and unscheduled [6]. Also compounding the user dissatisfaction is the increasing price of electrical energy.

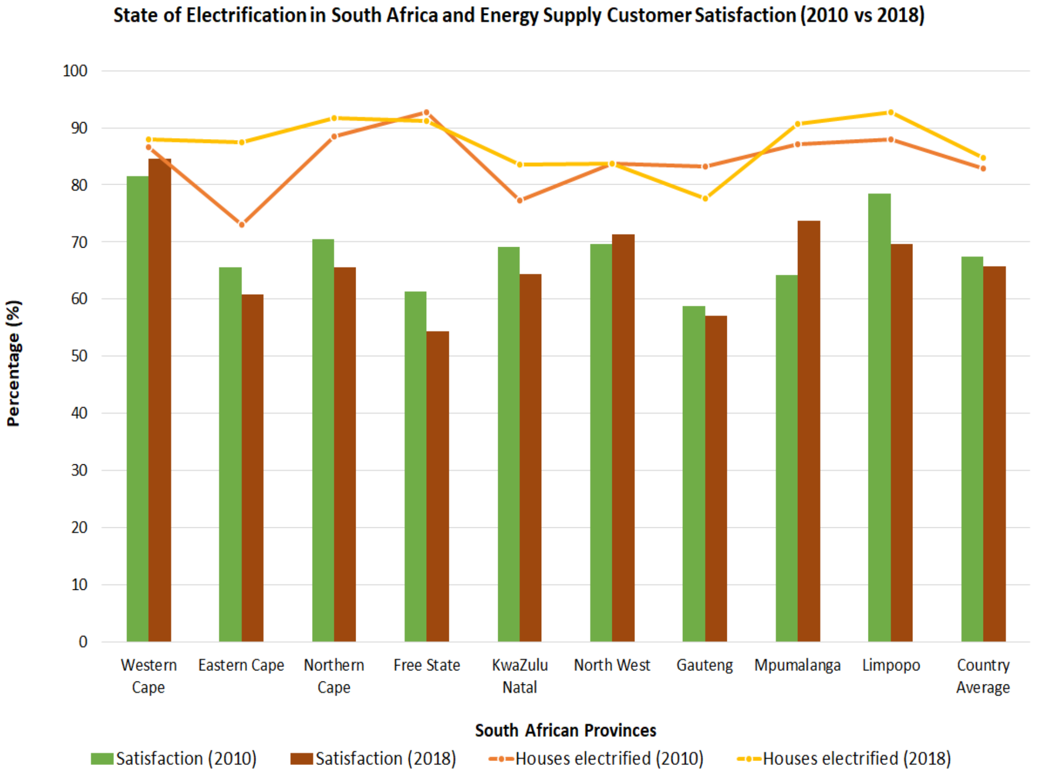


Figure 1. South Africa electrification status and overall customer satisfaction (generated from data presented in the South African Census report [4,5]).

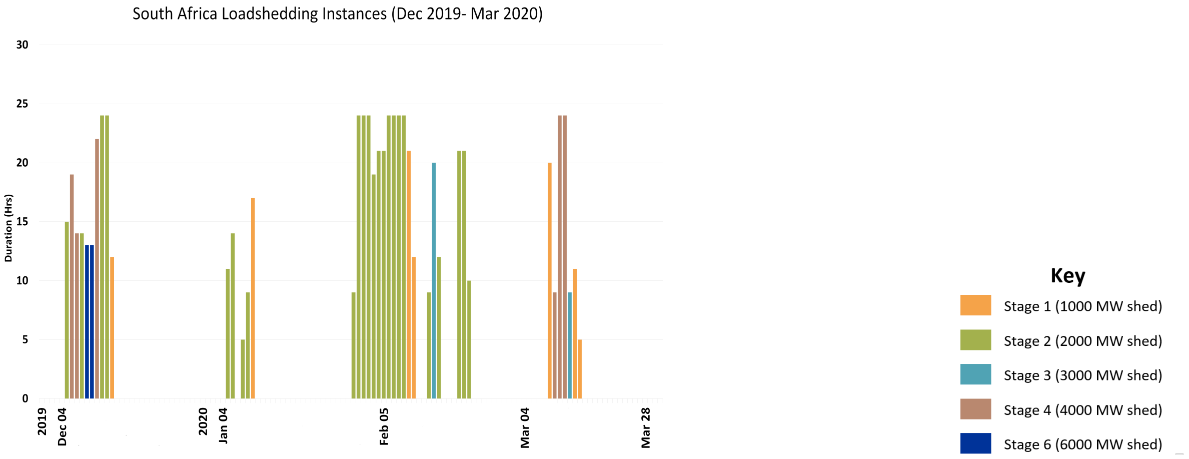
Unlike Western economies, South Africa faces four major problems as part of the electricity crisis: continuing efforts towards electrifying the country, especially in more rural areas, improving the quality and reliability of electricity supplied to decrease its citizen's reliance on alternative generation structures, incorporating more renewable generation sources to decrease the country's reliance on a depleting coal supply for electricity generation and upgrading generation and distribution infrastructure in order to minimise maintenance related outages and accidents. In addition, the national energy provider, Eskom, has been faced with non-payment and non-compliance by municipalities on outstanding electricity bills [7,8]. As a result, power is being cut to defaulting municipalities and wards, leaving many without electricity. Gross mismanagement, corruption, looting, rising debt, failing infrastructure and a bloated workforce has crippled the power supplier and Eskom has been forced to lobby both government and the National Energy Regulator in recent years for funds to try and keep afloat [1,9]. This has led to an increasingly unstable grid as regular, scheduled maintenance is unable to be carried out.

A study by the Energy Centre at the Council for Scientific and Industrial Research found that the South African national energy provider scheduled blackouts totalling 858 h over the course of 2015. This translated to 1325 GWh of load shed over the course of these blackouts [10]. Integrating the new Medupi power station provided increased generation capacity; resulting in two years of uninterrupted energy supply. The last quarter of 2018 saw the return of scheduled loadshedding blackouts. The first quarter of 2019 experiencing 769 GWh of loadshedding over 272 h [10]. The last two months of 2019 saw the re-introduction of loadshedding [11,12], which continued until March 2020. Figure 2a gives an illustration of the loadshedding incidents that occurred between December 2019 and March 2020, the bar chart colours correlating with the loadshedding stages given in Figure 2b. It can be seen that in the 122 day period in which loadshedding activities were re-implemented, interrupted electricity supply was experienced for 42 days, with the duration of interrupted supply increasing dependent on the loadshedding level in place. The year 2019 also saw a steep increase in the price of electricity going from 118.56 c/KWh in 2018 to 153.90 c/KWh in 2019 [13,14]. As a result, more South Africans have been turning to other, fossil fuel based methods of energy generation. This further contributes towards air pollution and poses the danger of accidental household fires, child poisonings or carbon-monoxide related deaths [15].

With the unpredictability of the grid stability and the large disruption caused by loadshedding, market interest in independent generation would be increased, making a smart microgrid solution an attractive alternative. The ability to self-regulate electricity use along with the incentive towards revenue generation from energy trade and utilise cleaner energy sources means that a moderate to high initial buy-in could be achieved amongst the middle to upper middle class. Careful planning would be needed to implement microgrid solutions in the more rural areas of South Africa to get buy-in from the poorer communities.

Examples of typical rural homesteads are pictured in Figure 3. With the absence of centralised control and existing electricity infrastructure, such communities would need to be able to transact securely and independently amongst themselves. A distributed, decentralised marketplace would also facilitate transparent, real time price and demand forecasting. Contract-based transactions would need to provide non-repudiation between user agents. Automated financial settlement of energy trades would need to allow for the integration and participation of mobile, inanimate trade agents such as electric vehicles and energy generation farms. Users should be provided assurance that market data is unmodified and authenticated. The privacy of customer, market and corporate information should be maintained. Information delivery is also required to be timely and reliable [16].

One technology proposed for microgrid market operations are distributed ledger technologies (DLTs). These are immutable, distributed digital records updated by network participants and cryptographically sealed to ensure the authenticity of records and network transactions. The authenticity of energy trade transactions is guaranteed owing to public key cryptography and digital signatures. Trust establishment moves away from the definition of individual trust into community trust. Consensus establishment mechanisms ensure that network transactions are easily traceable through the ledger history owing to hashed transaction IDs. These native properties therefore make DLTs an attractive first solution for the secure energy market.



(**a**) (**b**)

Figure 2. Loadshedding incidents from December 2019 to March 2020 (graph generated from data collated from public announcements issued by the national energy provider, Eskom). (**a**) Outages from December 2019 to March 2020; (**b**) Key for implemented loadshedding stages.

(**a**) (**b**)

Figure 3. Rural S. African homesteads without electricity services. (**a**) Paulpietersburg, KwaZulu Natal, South Africa. (**b**) eNgojini, Eastern Cape, South Africa.

Despite the inherent advantages, DLT-enabled networks do face a number of challenges. Decreasing transaction approval rates, increasing mining costs, restricted network size, high energy and processing requirements are often seen in blockchain networks. DLT solutions also face a variety of security concerns, such as protection against 51% of attacks. The smart microgrid requires strict availability and latency deadlines and utilises low power, resource-constrained devices.

This work aims to establish the feasibility of a DLT-enabled secure, transactive energy market by highlighting how DLT solutions perform in the IIoT context when implemented on a low power device intended for use at the network edge and where memory, processing and energy resources are highly limited. The experiments conducted deploy a simple transaction-based smart contract onto a constrained edge device for the purpose of monitoring the operational state of the single device in two different operational states: when acting as a dedicated DLT node and when acting on both DLT and edge network operations in parallel. By conducting the performance analysis after enabling DLT and smart contract functionality on a commonly used edge device and evaluating the results against the operational restrictions required of an IIoT real-time network, this work aims to identify areas in which DLT would not perform adequately in its native state thus highlighting the incompleteness of DLTs as a solution for the secure energy marketplace. From the experiment results and findings, this feasibility study aims to:

- Propose design considerations for the secure energy marketplace that is to be part of the DLT-IIoT enabled smart microgrid community.
- Establish a set of operational baseline performance metrics for a DLT-enabled edge node, which have been subject to the restrictions and deadlines seen in the IIoT network context.
- Identify security vulnerabilities that would need to be protected against within the design of the energy marketplace.
- Highlight the socio-economic limitations of rural smart microgrid deployments including infrastructure and telecommunications limitations.

This work expands on the foundation work conducted in [17], providing more detail on the experiments conducted in order to determine the operational efficiency of an Ethereum-enabled Raspberry Pi 3 by considering the CPU, RAM utilisation, execution time, core temperature changes, block mining times and power consumption of the device when executing smart contract transactions. These results are then compared to the theoretical performance estimations expected of an Ethereum node. A more in-depth analysis of the performance discrepancies seen between the Raspberry Pi and the theoretical estimations are highlighted in addition to possible causes for the discrepancies. This work also quantifies the possible interference that Ethereum operations may introduce in other IIoT processes by determining and comparing the execution time and core temperature changes to the Raspberry Pi when a processor-intensive program is run solo and concurrently with smart contract transactions. In establishing these baseline performance metrics, a comparison of solutions intended to improve DLT-IIoT efficiency can be conducted. Additionally, these results could be used in the emulation and process design of larger scale microgrid networks, which implement DLTs as part of the secure, energy trade market. The socio-economic restrictions that may hamper the deployment of the DLT-enabled smart microgrid are analysed and possible security vulnerabilities and restrictions are highlighted.

This paper is organised as follows: Section 2 provides a brief background of work that has been conducted on utilising DLTs in the smart microgrid context. Section 3 gives a brief overview of the proposed smart microgrid community model and discusses the theoretical expectations and limitations of the chosen DLT algorithm while Section 4 introduces the experiment methodology used in the performance evaluation and stress testing of the Raspberry Pi DLT node and its blockchain network. The experiment results are presented in Sections 5 and 6 discuses the suitability of DLTs as a solution for the secure transactive market when implemented at the IIoT network edge. Section 7 concludes the paper and provides a brief insight into future research directions.

2. Related Work: DLT-Enabled Industrial IoT Solutions

Smart microgrids represent a distributed, interconnected network of generation and storage loads. These loads are located at the grid edge and are capable of both grid connected and island modes of operation. Loads are managed by a microgrid controller, which presents the distributed network as a single, controllable load on the main grid [18]. Smart microgrid implementations aim to:

- Improve power supply reliability,
- Provide real time consumption and generation data for grid optimisation,
- Reduce dependencies on main generation sources,
- Reduce the carbon footprint of the energy sector, and
- Provide pricing flexibility to end users [19–21].

Fundamentally, the smart microgrid must meet numerous architectural goals, as described in [22]. As a highly critical and regulated application space, incorporating ICT with physical processes should not expose the grid to malicious actors. Appropriate security mechanisms should enable defence against “unauthorised intrusion, access or use of physical and cyber assets” [22]. Grid operations should also protect and obscure customer transactions and personal identifiable information at all levels of the network. Such mechanisms should also ensure not to disrupt normal grid operations. According to the *Guidelines for Smart Grid Cybersecurity* [16], security mechanisms for the smart microgrid should ensure that:

- A high level of availability is maintained as the primary concern,
- Information integrity and confidentiality of customer and grid information is provided [16].

With its history in cryptocurrency, DLTs enable both normal transactions and micro-transactions. This enables accurate partial purchases of energy according to each end users' specific energy deficit. The real time energy state can be updated concurrently with the ledger, allowing for accurate representation of energy consumption behaviour. This is essential for both demand and energy forecasting. The decentralised and distributed nature of the ledger means that network participants have a local, up to date copy of the ledger. Combined with a well formulated smart contract, this could allow for the automated sale and purchasing of energy by low power, embedded IIoT-device participants. Actors such as electric vehicles (EVs) and smart meters/controllers could independently trade energy after an initial wallet and network ID is established for them. This makes energy management processes more organic as conscientious decision making can be left to the microgrid controllers in each household. Real time adjustments to the household energy consumption could be made by autonomously switching off unneeded devices. The trade of electrical energy could be made once certain generation thresholds have been surpassed and projected household consumption demands have been met.

DLTs were first seen within the cryptocurrency application space with the advent of Bitcoin in 2008. The technology branched off from existing distributed databases, aiming to solve their problems with inconsistency, unreliability and poor node-to-node communications [23]. DLTs, however, are rapidly moving away from solely providing a method of currency trade into other application spaces. Research has been conducted towards the incorporation of DLTs into various areas of the smart microgrid architecture.

DLT-enabled, Vehicle-to-Grid energy trade is being intensively explored by various authors. As mobile microgrid actors, hybrid EVs could serve as an additional means to transport emergency power between microgrid networks using their battery banks and their ability to charge each time a driver breaks the car. Zhou et al. [24] propose a vehicle-to-grid framework for the Internet of Energy (IoE). The framework utilises a consortium blockchain and edge computing technologies to provide a distributed security mechanism for local energy aggregators (LEAGS). This is then used to create, verify and propagate energy sales between electric vehicles and energy service providers at minimum operational costs [24]. Wang et al. [25] propose a contract-based charging scheme for electric vehicles in the IoE that is managed and secured by a consortium blockchain. Yu et al. [26] explore electric vehicles, in the role of mobile energy transporters, as a means of energy demand response to various smart microgrid districts, resulting in the creation of mobile Vehicle-to-Grid energy networks. Energy trading platforms for microgrid smart homes are explored in [27] by Kang et al. in which a private blockchain network is established for IoT devices found within a residential building. A home miner monitors the energy usage of devices in-house and manages an energy trading process between residences of the same neighbourhood through the use of smart contract facilities [27].

A common concern raised by the authors in [24,25,27] is the large processing and high energy requirements for successfully running blockchain technology. As one of the core implementing technologies, devices used within the IIoT are typically constrained in their hardware capabilities and energy resources. These devices also operate in mission critical environments with high availability requirements. Reliability, security, privacy and low latency are prioritised for large data exchange communications [28–33]. An investigation is thus required into the impact that the processor-intensive DLT operations could have on device operations. A performance analysis of energy transactions confirmation times was simulated in [34] for 50 pairs of IIoT nodes. The latency introduced by multiple blockchain transactions and the mining power required for IoT networks was investigated in [35] for a variety of workloads. In extending this body of research, the authors take a more focused approach by investigating the expected performance and operational state of an individual, DLT-enabled node, under the stricter availability conditions for the smart microgrid. The shortfalls of native DLT implementations in mission-critical IIoT applications could be determined. In addition, areas in which further design considerations may be required, whether architecturally—in terms of DLT operations distribution—or algorithmically—in the form of optimised consensus mechanisms—could be determined as part of further efforts towards realising the DLT-enabled, secure, transactive energy market.

3. The DLT-IIoT Enabled Smart Microgrid

Figure 4 gives a highly simplified view of how the DLT-enabled smart microgrid community could be realised in the emerging economy. Customer demand for electrical energy is often scattered across relatively wide areas, with supply being handled by centralised generation stations and a high voltage transmission grid [36]. Topological and geographic challenges make it difficult to easily and quickly connect new communities to the centralised grid. The proposed microgrid architecture aims to facilitate the faster realisation of universal electrification by employing cost-competitive distributed renewable energy generation sources to form a series of interconnected microgrids [36].

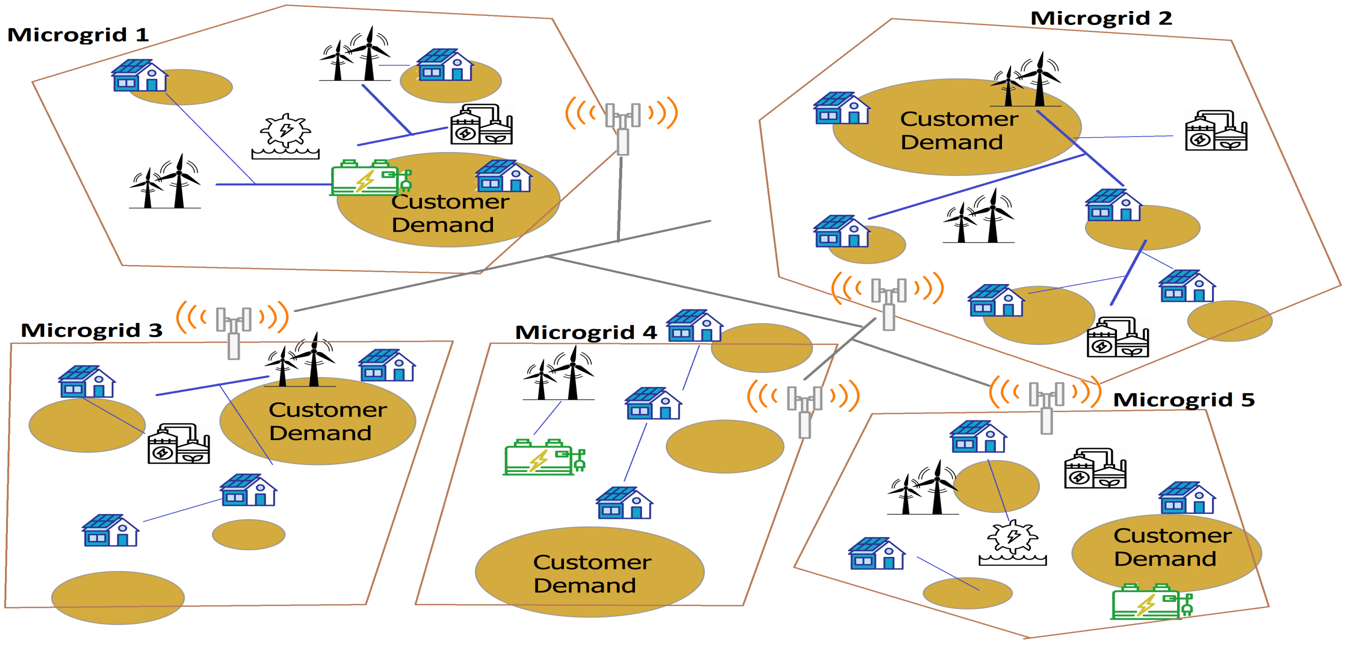
The model architecture considers a small, rural community of 50–100 people in government subsidised housing, forming around 25–50 neighbouring household nodes. Each household would have a renewable generation source—such as solar panels, solar geysers or wind turbines—as the main electrical energy supply for space heating, cooking, refrigeration, lighting and entertainment purposes in conjunction with an energy storage device.

Energy management operations would be handled by a low power, IIoT edge device acting as a DLT-enabled controller unit, which may take the form of a new-generation smart meter. Combined with a variety of sensing technologies, adjustments to the household consumption would be made by eliminating unneeded devices from the active devices log while energy supply from generation sources is kept at safe operational levels for household consumption or redirected to energy storage devices to ensure an adequate backup supply is maintained.

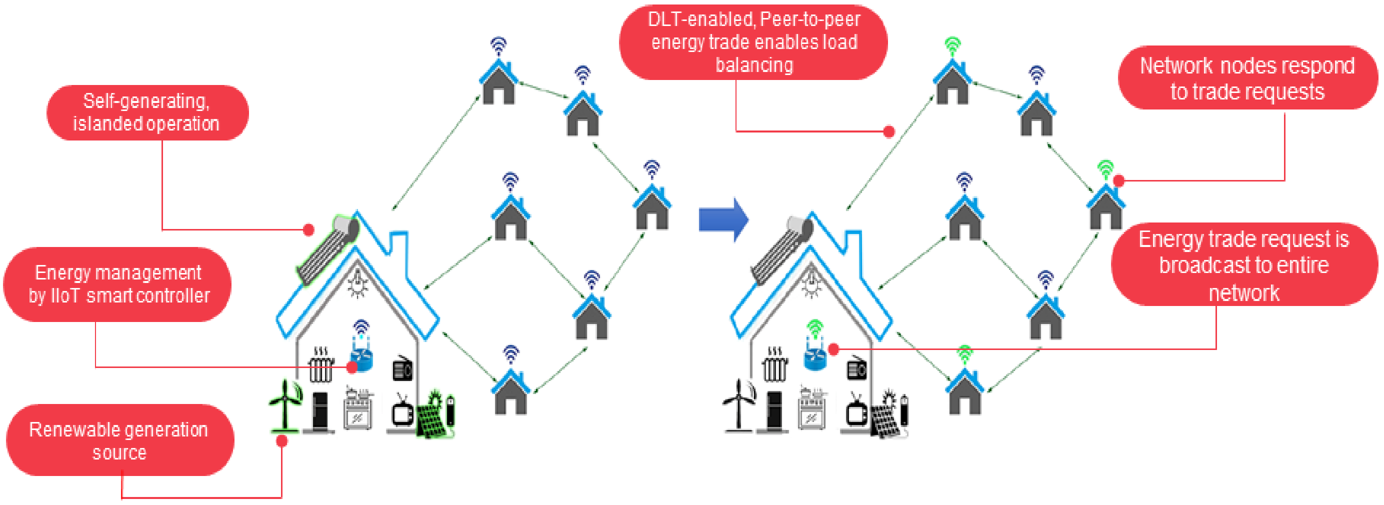
Mining operations, inter- and intra- community energy trade transactions shall be managed by a base station acting in the role of a community node. The base station device shall be a high power IIoT device with more processing and memory resources available to allow for information exchanges at higher levels of the network architecture. The base station shall be equipped with solar generation and storage capacity to minimise additional strain on the microgrid generation capacity.

Should a household register a supply deficit or surplus, the edge controller node would connect to the local microgrid network with a transaction request. With the high cost of mobile data in South Africa, interrupted mobile network connection and 3G download speeds between 4.2 and 6.4 Mbps [37], microgrid nodes would be limited to approximately 50 distinct transactions daily, with fast completion rates required for buying and selling activities. With an established participant identity on the DLT network and wallet and smart contract processing capabilities, the controller would facilitate the sale or purchase of energy from neighbouring nodes in the microgrid dependent on the real time market price and available energy. Households would be able to supplement or contribute towards meeting immediate energy needs—and thus maintaining a stable, continued supply throughout the microgrid—while ensuring no wastage of generated electrical energy. In having a financial incentive for the energy trade, microgrid participants are more eager to contribute towards a community-like generation and supply network.

The community smart microgrid DLT would be best serviced by a permissioned, public ledger structure as seen in Proof of Stake based consensus protocols. This would allow for easy joining and transacting between nodes within a neighbourhood network while preventing joins from outside the community [38]. To facilitate trade between communities, each neighbourhood would be converted into a single node based upon their municipal demarcation zones. These zone nodes could trade between each other in the same manner as the smaller community microgrids without requiring individual peer-to-peer sales between homesteads.



(**a**)


Self-generating, islanded operation
Energy management by IIoT smart controller
Renewable generation source
DLT-enabled, Peer-to-peer energy trade enables load balancing
Network nodes respond to trade requests
Energy trade request is broadcast to entire network

(**b**)

Figure 4. The distributed ledger technologies (DLT)-enabled microgrid community. (**a**) Proposed microgrid architecture for rural network deployments [36]; (**b**) Operation of the DLT-IIoT microgrid can be broken into two main stages: isolated, self generation and DLT-enabled energy trading (adapted from [17], ©2020 IEEE).

To enable the sharing of electrical energy and flexible, real time pricing, a secure energy marketplace in a smart microgrid would need to be established. Sufficient facilities need to be implemented to ensure marketplace honesty. Some of the safeguards required would include:

- That users are selling resources that are available,
- That resources are fully owned by the seller,
- That users are protected in any energy trade transactions that they participate in.

Success of the secure energy marketplace is also dependent on other fundamental characteristics. Historic market transactions would need to be readily available in a distributed, decentralised manner. Rural communities in South Africa are isolated from electricity distribution sources and the distance and land topology make connecting communities difficult.

With its support for consortium-based networks, an existing code base for IIoT platforms and smart contracting facility, Ethereum was chosen as the first DLT implementation to be evaluated for the smart microgrid context. Currently, single Ethereum nodes are benchmarked at being able to support 1000–2000 transactions per second with a consortium handling hundreds of transactions per second [39]. This figure, however, could vary owing to the computational limitations of IIoT devices and the bandwidth limitations of the smart microgrid network. As it stands, a single Ethereum transaction requires [39]:

- One elliptic curve signature validation,
- A minimum of three state updates with the possibility of 10 state updates. This directly corresponds to 3–10 state updates of Ethereum's Merkle tree data structure. Merkle tree synchronisation has, on average, a complexity of $O(\log_2(n))$ with a worst case of $O(n)$ [40].
- $+/-$ 15 to 100 rounds of serialisation and hashing operations as a result of the Merkle tree synchronisation.
- Approximately 15–100 updates to the Ethereum database
- At least one virtual machine execution in the case of smart contract calls.

In the end, an Ethereum node processes a minimum of 35 operations on a single transaction, with the worst case possibility of 212 operations on a single transaction. Concurrently with this, IIoT devices would also be required to maintain other edge processing operations facilitating the power management policy of each home node. Thus, sufficient resources would still need to be made available during the execution of Ethereum operations to ensure no interference is introduced by DLT operations.

Other features within the native design of Ethereum could also pose some difficulties for IIoT operations. The number of confirmations needed per transaction, the network congestion, block size and the gas price allocated to the transaction impacts the speed at which transactions are added to a block for mining [41]. Prior to being added to a block, pending transactions are added to a Mem pool from which miners are able to pick transactions to bundle into their blocks [41]. At the time of writing, Ethereum blocks have an average limit size of 9.9 million gas per block [42]. Transaction confirmation speeds can be adjusted depending on the gas price declared [41]. In order to manage the speed at which blocks are input into the Blockchain, Ethereum adjusts the difficulty of the PoW calculation using the *block_time* equation and a difficulty bomb [43]. As this is based on computational power available in the network, the block time could vary slightly from the expected block time as the difficulty bomb is adjusted per block [43].

Apart from the network state, the resources available on an Ethereum node also serves to impact the efficiency of the node sync time. In 2018, the minimum resources required for a full Ethereum node on the mainnet were investigated by the author in [44]. The author found that for synchronisation purposes, disk throughput had a higher impact on sync time than the CPU. This is important for IIoT devices where resources are more constrained and have lower throughput than traditional PC devices.

Table 1 gives an overview of the state of the Ethereum mainnet at the time of writing. The transaction processing time and the rate at which blocks are added to the mainnet are much longer than would be acceptable for the near real-time requirements of the energy market. These times could also increase given the more limited resources available on IIoT devices. In order to determine whether Ethereum could feasibly be run in an IIoT context, a private testnet independent of the mainnet is required in order to better control the rate of network growth and establish the minimum transaction and block times, which could be seen from an IIoT configuration.

Table 1. State of Ethereum mainnet.

Number of mainnet blocks	9,778,659 blocks [45]
Standard 1-to-1 gas price (mainnet)	21,000 gas [46]
Number of confirmations per transaction	30 [47]
Transaction processing time	6 min [47]
Ave. Block time (January 2020)	13.0 s [45]
Ave. Difficulty (March 2019)	2.2210 PH [42]
Ave. Transactions per day (March 2020)	+/−700,000 [48]

4. Evaluating Ethereum in the Smart Microgrid Context

Within the microgrid architecture proposed in Figure 4, low power, resource constrained edge devices are to be utilised within the smart meter controller design alongside smart plugs and other low level sensing devices. The controllers are used to manage household consumption levels by maintaining an active appliances log and initiating and participating in energy trade operations through smart contracting facilities. As processing, memory and energy resources on edge devices are highly constrained, performance testing experiments were designed using known techniques in order to determine the feasibility of running DLT algorithms on a low power platform. Initially, the Cortex M class devices were considered; however, compatibility issues with the MCU architectures led to switching to another commonly utilised IIoT edge device.

Thus, to provide the best chance of a successful deployment on a low power, edge device platform, the Raspberry Pi 3 was chosen; owing to the 64-bit ARMv7 architecture's out-of-the-box compatibility with a large variety of DLT implementations, and its more powerful processing and memory specification as compared to other SoCs intended for IoT and IIoT use [49]. Given that an operating system (OS) is required to run most implementations of DLTs, a new image of the Raspbian Stretch 4.4 OS was flashed onto a 32GB UHS-I microSD card, providing a maximum read speed of 80 MB/s and a minimum write speed of 10 MB/s. To get the best performance from the node, the full SD card memory was made available to the OS and a 1024 MB swap file was created. With a compatible port available for the Raspberry Pi 3 and native smart contract support, Ethereum was chosen to implement the private test network used for experimentation. The Ethereum code was pulled, unaltered, directly from the publicly available git repository and was installed and implemented according to the instructions provided in [50] using the Go implementation client *Geth*. The installation allows the Pi to be able to create a wallet, send and receive transactions from the blockchain and was deployed in *fast sync* mode.

Other network activities, such as mining and the compilation of new network blocks, are not possible on IIoT edge devices, owing to the hardware and processing restrictions. As per the proposed microgrid architecture, these activities shall be handled by a base station acting in the role of a community node. In this experiment, a Dell Optiplex 9020 PC— with an Intel i7-4790 vPro CPU, 8GB RAM and running Windows 10 Enterprise— was used to act as the base station miner node; however, in future designs, the base station shall be implemented on a high power IIoT platform such as the Smartfusion 2. The smart contract used to implement simple transaction operations was deployed onto the private test network using the Truffle development environment on the miner PC and Node.js on the Raspberry Pi.

4.1. Performance Evaluation of the Raspberry Pi Ethereum Node

To provide a broad overview on the operating performance of the Raspberry Pi as it executed native Ethereum, a variety of metrics were considered:

- Transaction execution time: the time taken by the node to complete a transaction requested by the smart contract.
- Transaction power consumption: the power consumed by the node during the execution of the transaction.
- Node CPU and memory consumption: the use of RAM and CPU resources by the node in various operational states.
- Node core temperature: the temperature of the IIoT SoC for varying operational states.

To evaluate the performance, a simple smart contract was deployed. According to the contract specifications, the Raspberry Pi node was to transmit 'send' transactions into the network over a set period of time by transferring one ether to the wallet address of the miner node. For this set of experiments, 1 and 2 min intervals were selected as the total time period in deference to the overheating potential of the Pi's processor. The time between the smart contract calls was adjusted in various rounds to test a variety of time intervals that cover normal edge node operations in a wake/sleep pattern (transaction send every 10 s) and real time edge node operation (transaction send every 1 ms). To determine the execution time and power consumption associated with the Ethereum transaction, a GPIO pin toggling method was implemented within the contract in addition to the transaction methods. Starting with the pin in a LOW state, at the start of each 'send' transaction the pin was pulled into a HIGH state. Once the transaction had completed, identified by a decrease in the wallet balance, the GPIO pin was pulled back to its starting LOW position. The initialised GPIO pin was connected to a Rigol DS1102D oscilloscope across a 1.3 Ω resistor to measure the width time and voltage amplitude of the resultant square waveform. Figure 5 gives an illustration of the setup for the Raspberry Pi and oscilloscope with the resultant waveform shown in Figure 6b.

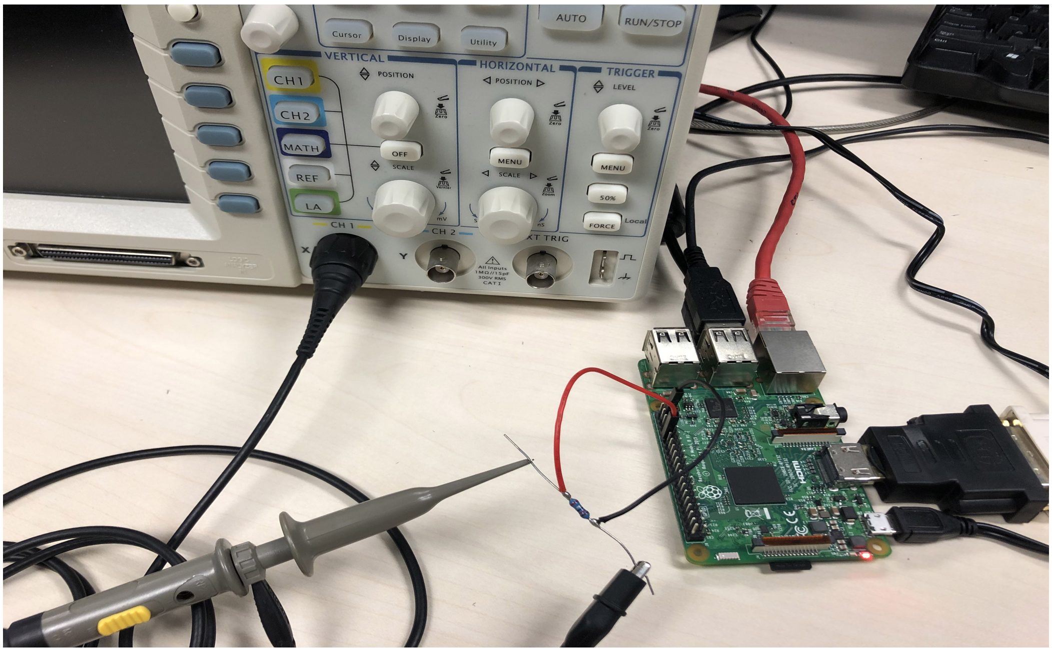
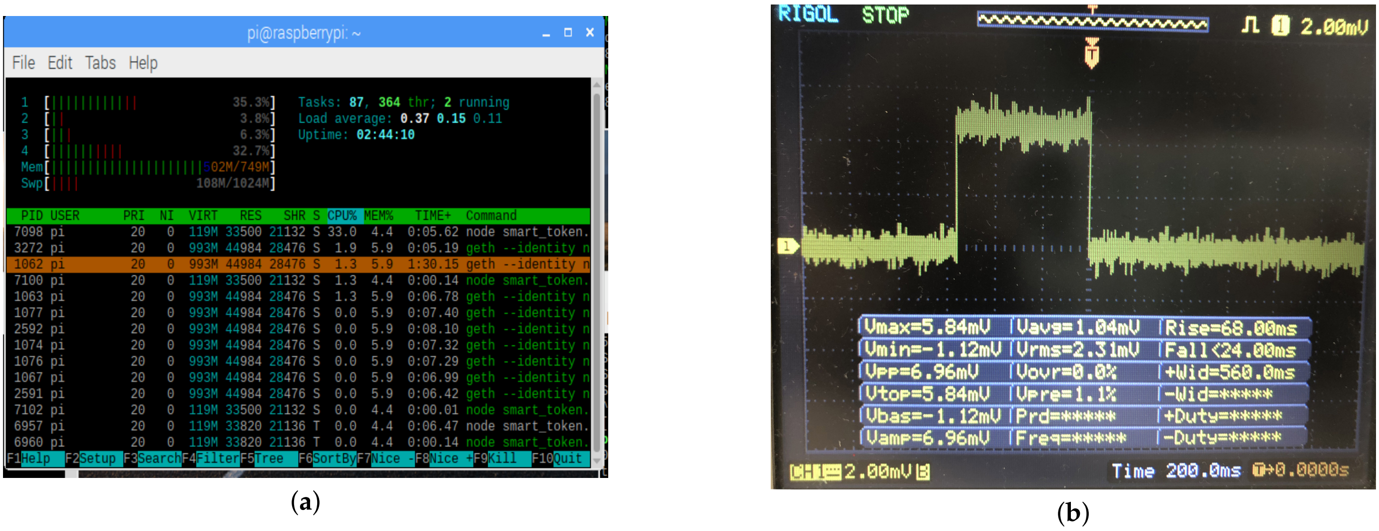


Figure 5. Raspberry Pi Ethereum experiment setup.



```
  1  [|||||||||||||               35.3%]   Tasks: 87, 364 thr; 2 running
  2  [||                           3.8%]   Load average: 0.37 0.15 0.11
  3  [|||                          6.3%]   Uptime: 02:44:10
  4  [||||||||||                  32.7%]
  Mem[|||||||||||||||||||||502M/749M]
  Swp[||||                   108M/1024M]

  PID USER      PRI  NI  VIRT   RES   SHR S CPU% MEM%   TIME+  Command
 7098 pi         20   0  119M 33500 21132 S 33.0  4.4  0:05.62 node smart_token.
 3272 pi         20   0  993M 44984 28476 S  1.9  5.9  0:05.19 geth --identity n
 1062 pi         20   0  993M 44984 28476 S  1.3  5.9  1:30.15 geth --identity n
 7100 pi         20   0  119M 33500 21132 S  1.3  4.4  0:00.14 node smart_token.
 1063 pi         20   0  993M 44984 28476 S  1.3  5.9  0:06.78 geth --identity n
 1077 pi         20   0  993M 44984 28476 S  0.0  5.9  0:07.40 geth --identity n
 2592 pi         20   0  993M 44984 28476 S  0.0  5.9  0:08.10 geth --identity n
 1074 pi         20   0  993M 44984 28476 S  0.0  5.9  0:07.32 geth --identity n
 1076 pi         20   0  993M 44984 28476 S  0.0  5.9  0:07.29 geth --identity n
 1067 pi         20   0  993M 44984 28476 S  0.0  5.9  0:06.99 geth --identity n
 2591 pi         20   0  993M 44984 28476 S  0.0  5.9  0:06.42 geth --identity n
 7102 pi         20   0  119M 33500 21132 S  0.0  4.4  0:00.01 node smart_token.
 6957 pi         20   0  119M 33820 21136 T  0.0  4.4  0:06.47 node smart_token.
 6960 pi         20   0  119M 33820 21136 T  0.0  4.4  0:00.14 node smart_token.
F1Help  F2Setup F3SearchF4FilterF5Tree  F6SortByF7Nice -F8Nice +F9Kill  F10Quit
```

(**a**)

```
Vmax=5.84mV   |Vavg=1.04mV |Rise=68.00ms
Vmin=-1.12mV  |Vrms=2.31mV |Fall<24.00ms
Vpp=6.96mV    |Vovr=0.0%   |+Wid=560.0ms
Vtop=5.84mV   |Vpre=1.1%   |-Wid=*****
Vbas=-1.12mV  |Prd=*****   |+Duty=*****
Vamp=6.96mV   |Freq=*****  |-Duty=*****
CH1 2.00mV B                  Time 200.0ms  T→0.0000s
```

(**b**)

Figure 6. Performance testing experiment captures. (**a**) Htop capture of CPU and RAM memory consumption; (**b**) Execution time and voltage drop waveform for Ethereum transaction.

The core temperature of the node was measured using the built-in Raspberry *measure_temp* command and the on-board CPU temperature sensor [51] reading at one second intervals over the contract execution period. This was to determine the heating effect of the constant processing of the DLT transactions on the CPU, as the operating temperature would not only affect the design of the edge node enclosure but also exceeding the safe, recommended operating temperature of the SoC could lead to damage to the components, compromised execution and a shortened node lifespan. Temperature readings were taken over a 60 s period in four different states: right after powering up the Raspberry Pi, labelled as Startup, while the node is running the Ethereum code and has an attached Geth instance but without executing any transactions, labelled as Idle Node, during contract execution with transactions being sent to the blockchain, labelled as Contract Execution, and right after the contract has finished execution for the active time period, labelled as After Contract Execution.

The use of memory and processor resources, as percentages of the total available RAM and CPU utilisation, respectively [52], were noted concurrently with the execution time experiments by running the Linux *htop* command from a new terminal. This output can be seen in Figure 6a. To differentiate the Ethereum processes from other background OS processes, the *htop* output was filtered to only show the processes related to Ethereum. For this metric, the four node states identified in the core temperature experiments were used. The results observed could be used in future works as a means to extrapolate possible node processor and memory resource behaviours over longer operation time periods and could aid in identifying a node's capacity to support and execute other edge operations concurrently with Ethereum transactions.

4.2. Stress Testing the Raspberry Pi Ethereum Node

With the move towards bringing processing activities closer to the network edge, nodes in the IIoT network will be expected to be able to run some pre-processing on sensor data prior to sending to their neighbour edge or fog nodes while still meeting throughput requirements that guarantee a high level of availability is maintained in the network. DLTs are known to be processing intensive activities, even in light node configurations; thus, their inclusion on edge devices have the potential of interfering with other processing activities and introducing latency into the network. To try and determine the effect that the inclusion of Ethereum could have on edge processing activities, a stress test was designed to have an additional, processing intensive program run in parallel during the Ethereum smart contract execution. A python script was written to have the value of π calculated to 500 places using the Archimedes' method, as described in [53].

The first 100 results were output to the terminal along with the core temperature reading at that point in time. The execution time of the script was given once the 100th result had been completed. The interference of the Ethereum operations would therefore be determined by changes in the script execution time as well as the final core temperature increases observed.

Once again, four different operation states were considered. As in the performance evaluation experiments, the Startup state and the Idle Node state were tested. In addition, two different contract execution time periods were run. The first state had the contract execute an Ethereum transaction every 10 s, representing operations that do not have real-time or near real time restrictions. This was denoted as Contract Execution (10 s). The second state had the contract execute an Ethereum transaction every 1 ms, representing real-time or near-real time operations, and was denoted as Contract Execution (1 ms).

4.3. Updating the Blockchain

In addition to evaluating the Raspberry's node performance, the fees and block processing time associated with each experiment run was recorded to determine the effect of the increasing network size, as a result of the transactions produced, on the mining cost of the resultant blocks as the effort required to verify transactions and solve the Proof of Work increased. Mining on the miner node was halted until the conclusion of each experiment was set. Once started, the mining output data for the resultant block was captured. This included the number of transactions within the block, the gas fee required for the block and the confirmation time for the block to be added to the ledger.

5. Results

The mean results for the performance evaluation, stress testing and mining experiments are presented in the following sections, rounded to three decimal points. Standard deviation was determined according to (1) and was also rounded to three decimal points. This was illustrated in Figures 7a,b and 8 as error bars on each mean result, with the value of the standard deviation result displayed above.

$$\sigma = \sqrt{\overline{X}} \;\; where \;\; \overline{X} = \frac{1}{N}\sum_{i=1}^{N} X_i \tag{1}$$

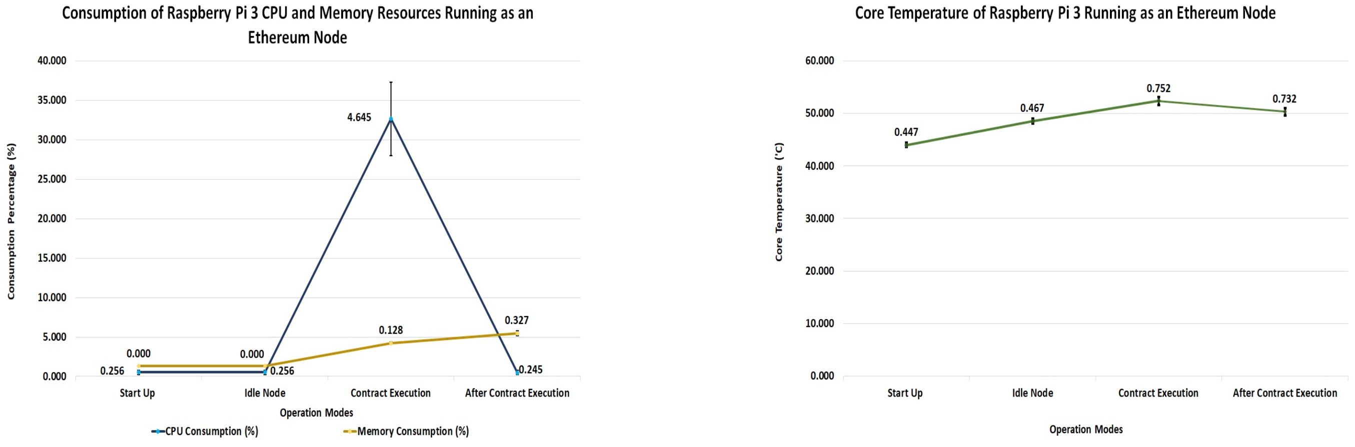
5.1. Performance Evaluation

At the conclusion of the performance evaluation experiments, the Raspberry Pi was observed completing an Ethereum transaction with an average time of 513.400 ms and an average deviation of ±18.904 ms. This is approximately 500 times longer than the benchmarked execution time of 1 transaction/ms for a single Ethereum node. In a later experiment, an average time of 10 min was observed for the generation of 1000 Ethereum transactions. This supports that, as noted in the Ethereum white paper, the computational limitations of a node would have a significant effect on its transaction processing time. The current consumed from the CPU (I_{MCU}) was calculated from the observed voltage drop (V_{MCU}) and the shunt resistor (R) according to Ohms Law. This was then used to determine the power consumed by the Raspberry using (2):

$$P_{MCU} = V_s \times I_{MCU}\; W \tag{2}$$

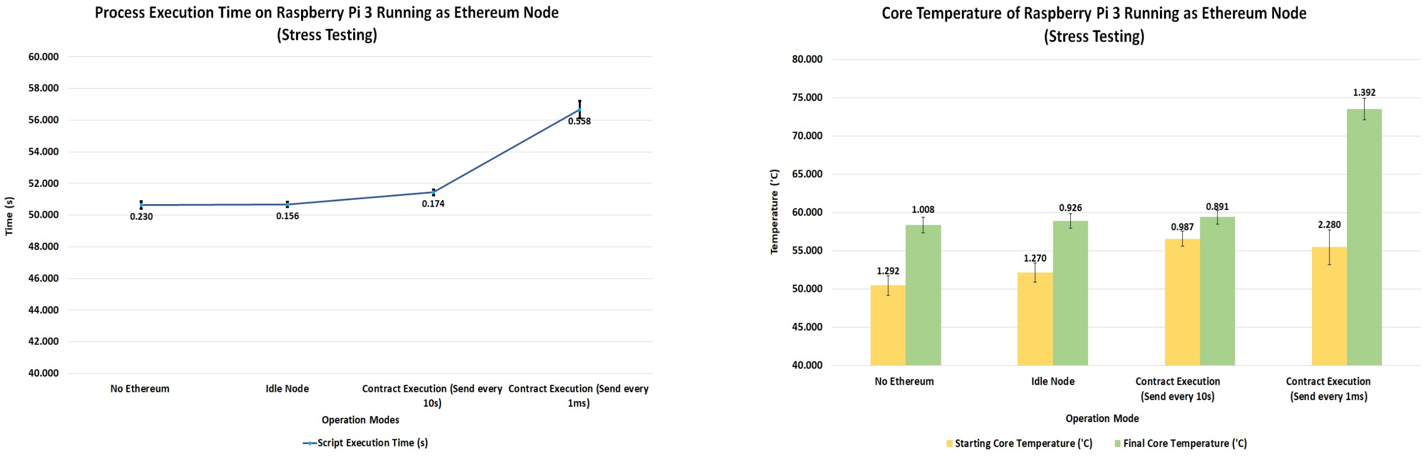
where the supply voltage to the Raspberry Pi V_s is given as 5 V. This result was then converted to milliwatts (mW). Subsequently, the average power consumed by an Ethereum transaction was observed to be 25.528 mW with a deviation of ±2.117 mW.

Over the course of the experiment, a general increase in the core temperature of the Raspberry was observed. Considering Figure 7b, one can see that the node in an idle state experienced an increase of 4.583 °C in the average temperature when compared to the average start up temperature of 44.000 °C. Contract execution lead to an 8.353 °C increase, as compared to Startup, with the temperature decreasing by 2.053 °C once the contract had concluded execution. Between the Idle Node state and the Contract Execution state, the Raspberry Pi experienced a 3.770 °C increase in temperature. While the increases in temperatures observed were significant, it was noted that the maximum recorded temperature of 53.800 °C for Contract Execution was still well below the recommended maximum operating temperature of 80 °C for the Raspberry Pi.



(**a**) (**b**)

Figure 7. Performance testing experiment captures. (**a**) Htop capture of CPU and RAM memory consumption; (**b**) Execution time and voltage drop waveform for Ethereum transaction.



(**a**) (**b**)

Figure 8. Stress testing results of Raspberry Pi Ethereum node: (**a**) Concurrently running Python script execution time. (**b**) Core temperature of Raspberry Pi node under stress test.

For the CPU, the Idle Node and After Contract Execution states were not observed as having a higher utilisation rate than the Startup state. The main utilisation of the CPU occurred during the execution of the Smart Contract for each transaction sent to the blockchain network. On average, a transaction consumed 32.663% of the CPU with a deviation of ±4.650%. With the contract requesting a transaction to be sent every 10 s over the 60 s period, the high utilisation was observed as lasting only for the total time taken by the Raspberry to complete the transaction before dropping back down to the Idle Node state utilisation.

The consumption of RAM memory showed a relatively steady increase between the states. Consumption remained constant in both the Startup and Idle Node state with it increasing to an average utilisation of 4.240% ± 0.128% during the Contract Execution state and 5.467% ± 0.327% during the After Contract Execution state. This showed that sufficient RAM would be available to other edge operations even with the node busy running Ethereum transactions in parallel.

5.2. Stress Testing

The stress tests conducted on the Raspberry Pi looked into determining the possible interference that Ethereum transactions could pose on other node operations. Setting the baseline operation, without Ethereum running on the Raspberry, the python script was able to execute in an average time of 50.65 s with a deviation of ±0.230 s. The average temperature at the end of this execution was observed to be 58.35 °C with a deviation of ±1.008 °C; an increase of 7.88 °C from the starting temperature. Running the script while the Raspberry was in the Idle Node state gave similar results to those observed for the Startup state; giving an average execution time of 50.68s and a deviation of ±0.156 s while the average final temperature increased by 6.63 °C from the starting temperature of 52.14 °C. Both execution time and temperature results of the Idle Node state were similar to the results observed for the Startup state thus no interference from Ethereum was observed.

Following the first two tests, the Raspberry's smart contract was altered to have the node send a transaction every 10 s, approximating normal edge node operations. In this configuration, along with the average execution time increasing by 0.80 s, the final temperature rose to 59.38 °C. With a higher final temperature and a nearly 1 s increase in execution time, a slight interference between the python script and Ethereum was observed for this state.

In the final test, the smart contract was again altered, this time having the Raspberry send a transaction every 1 ms as an approximation of near-real time or real-time operations. The largest changes in temperature and execution time were observed for this configuration, with an increase of 18.04 °C on the starting temperature of 55.44 °C and an increase of 6.03 s for the execution of the python script. In this state, significant interference occurred in the execution of the script.

5.3. Mining the Transactions

Over the course of the various node performance experiments, 67 transactions were sent to the miner, breaking down into 29 Execution Time transactions, 29 CPU/Temperature transactions and 9 configuration transactions. For the 67 experiment transactions, 2,023,617 gas was required, incurring a fee of 0.002023617 Ether. This amounts to a price of approximately 30,203.24 gas per transaction or approximately 0.00003020324 Ether per transaction. For this block, it took 9.98 ms for mining scheduling and after 1.22 s, all the transactions were sealed as the single block 2294 and successfully mined. The stress tests generated a total of 1877 transactions and required 57,105,848 gas, incurring a fee of 0.057105848 Ether. For this instance, the gas price per transaction was 30,424 gas or 0.000030424 Ether. This is an increase of 220.76 gas and approximately 0.00000022076 Ether. This resulting block was mined 541.69 ms after the mining process was started. Unusually, the block containing more transactions was mined faster than that containing few transactions, illustrating that a smaller number of transactions per block may not result in a faster mining time.

Considering other mining blocks generated during testing, the following was observed:

- A block containing 177 transactions incurred a fee of 0.005385048 Ether and was mined in 763.56 ms.
- A block containing 124 transactions incurred a fee of 0.003772576 Ether and was mined in 94.95 ms.
- A block containing 125 transactions incurred a fee of 0.003803 Ether and was mined in 423.76 ms.
- A block containing 104 transactions incurred a fee of 0.003164096 Ether and was mined in 1.12 s.

It could be seen that the time taken by the block mining process is highly variable, even though blocks with a larger number of transactions offer more financial incentive to be mined faster. While this is ideal to ensure that smaller blocks do not get stuck waiting endlessly to be mined, it also means that there would not be a definitive guarantee as to when transactions are added to the ledger history, and thus made observable to the entire IIoT network. With the high degree of variability observed for the transactions generated from the node performance test, additional testing was conducted in order to try and determine a possible trend in the mining times that could be used to approximate the possible ledger update time. Over several runs, the mining times for blocks containing only 1, 10, 100 or 1000 transactions were captured in order to determine whether the number of transactions within the block had an effect on the resultant mining time. Owing to the limitations of the Raspberry Pi's transaction send time, the number of transactions that could be physically generated for the mining experiments was limited to a maximum of 1000.

Figure 9 shows that averaging out the observed mining times for each block size gave very similar mining times despite the increasing number of transactions contained within each block; however, a significant spike was observed in the mining times for blocks containing 10 transactions. This further illustrates that a high degree of unpredictability could affect the expected mining time of microgrid transactions, subsequently affecting the availability of the recent transactions for real time price and demand forecasting. A line of best fit was generated for the mining time data and it was observed that this overestimated the possible mining times when compared to the physical results. It does, however, serve to provide a reasonable approximation for the possible mining times that could be expected for blocks containing the number of transactions that are beyond what the Raspberry Pi is physically capable of generating in a reasonable length of time.

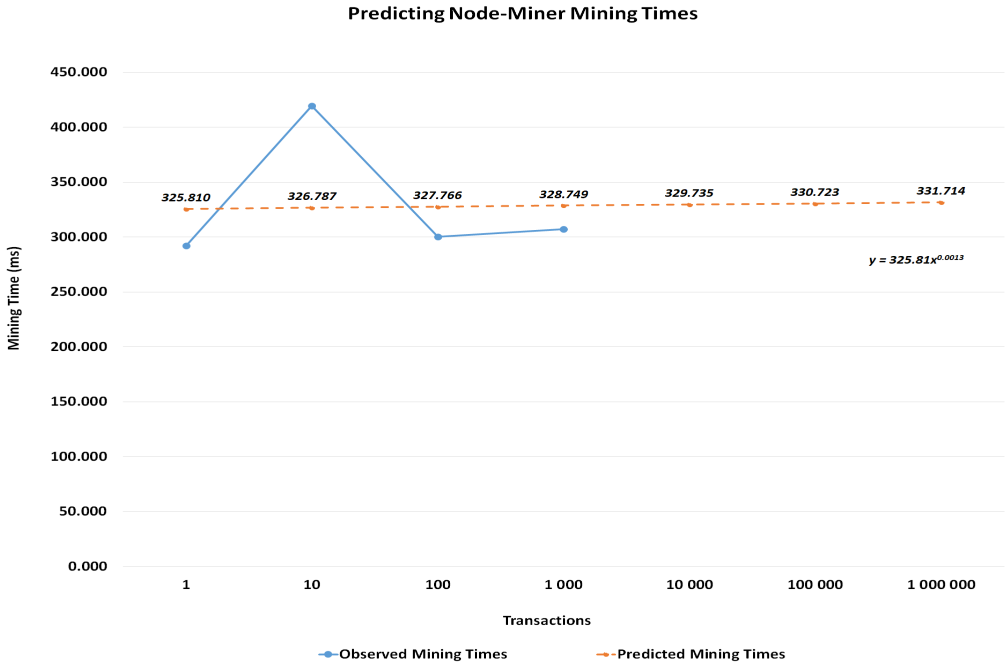


Figure 9. Mining time estimations for single node-miner pair.

6. Discussion

In order to be able to determine the feasibility of DLTs as a solution towards the energy trade marketplace, the operational performance of an IIoT edge device, the role of which would be household consumption management and autonomous energy trading through smart contracting facilities, was evaluated using known techniques in order to highlight areas in which DLTs may not perform adequately for IIoT applications. The following sections discuss the results generated from the conducted experiments in the contexts of the goals of the feasibility study to identify and propose design considerations for the energy marketplace, set a baseline of operational metrics for DLT-enabled edge device performance, identify the limitations of using DLTs in IIoT application spaces such as security vulnerabilities and the socio-economic and infrastructural climate of rural communities and highlight edge device platform compatibility restrictions that influence the design of the smart controller device.

6.1. DLT-Edge Node Performance

From the observed results, Ethereum performed well under the majority of the circumstances tested. Under normal node operation conditions, the DLT produced little interference to the regular operations. The observed power consumption for transactions was found as less than that observed for the low power Cortex M processors implementing standard, cryptographic algorithms [49]. The RAM memory consumption of the DLT was on par with that consumed by the Cortex M7 [49]. As a security mechanism for the IIoT edge, native Ethereum was found to be less power and RAM intensive than standard crypto algorithms. In the other areas assessed, and within the context of the smart microgrid, DLTs presented some problems. These would need to be handled as part of the development of a secure transactive market solution.

The execution times observed on the Raspberry Pi node were within the limits specified in [16] for specific smart microgrid scenarios including:

- Substation data,
- Non-critical equipment monitoring,
- Short term market pricing information,
- Meter reading,
- Long-term market pricing information and,
- Power quality information.

For real time or near real time operations, such as protective relaying and wide-area situational monitoring, the Raspberry Pi underperformed. It was unable to process Ethereum transactions within the time period specified by the standard, thus potentially introducing intolerable latencies into a real time or near real time network when handling Ethereum operation concurrently with IIoT operations. This was seen in the case of the contract requesting a transaction from the node every 1 ms—the minimum Ethereum node transaction generation rate. For the 60 s period in which the contract was active, 60,000 transactions should have been generated. For these transactions to be serviced, a minimum of 2,100,000 operations would have been required of the processing node. However, the Raspberry Pi was unable to meet the contract's demands and could only achieve an average execution time of 513.40 ms per transaction, with 177 transactions being generated before the time expired. This is equivalent to a minimum number of 6195 operations being processed within the 60 s time period. Considering the vast difference between the expected and actual results, it can be seen that 99.7% of the operations could not be processed by the Raspberry within the time limit, despite the CPU allocating significant processing power to transaction generation.

The Raspberry's performance under the stress test conditions presented another consideration for the implementation of DLTs at the IIoT edge. Normal operation conditions only resulted in a 1.58% increase in the script execution time, over the 60 s test period. Under the stress conditions, the temperature rose 5.04% from the starting temperature, with a maximum observed temperature of 61.20 °C. Over longer periods, the core temperature would continue to rise. Adequate cooling solutions would therefore be necessary as part of the enclosure design so to preserve the node lifespan.

The real-time operations of the node while running Ethereum transactions in parallel are of concern. An increase in core temperature of 32.53% was observed, with a maximum observed temperature at 75.80 °C. An increase of 11.90% in the execution time was also observed in the parallel operations. For such environments, architectural designs would need to consider areas where process off-loading could be beneficial towards meeting the operational time requirements.

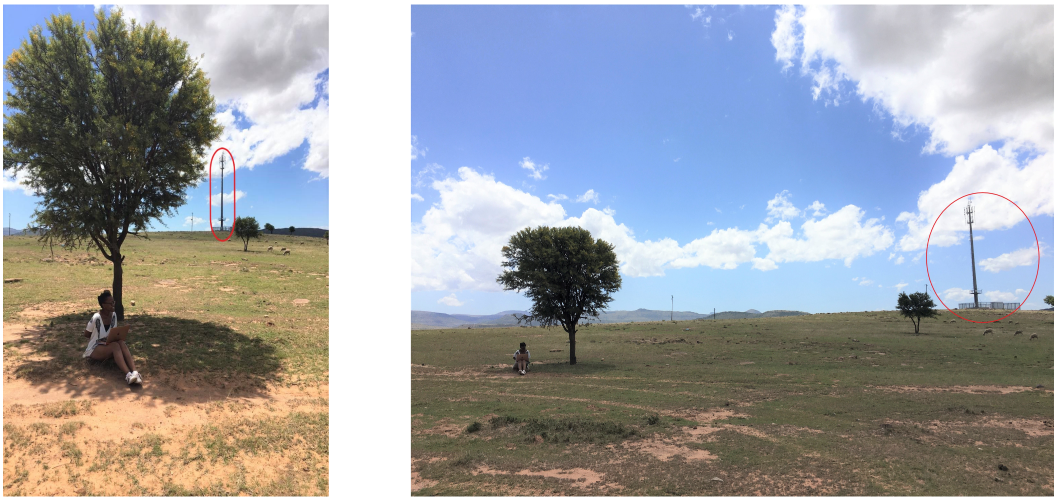
6.2. Socio-Economic Considerations for the DLT-Enabled Energy Market

In addition to the latency introduced, the network size and scalability of the Ethereum network could also pose a challenge for the secure transactive market place. The tests for this work were conducted on a small, private testnet with controlled network growth. In a live network, growth would occur much more rapidly. This is especially true for application spaces in which high levels of initial buy in are expected, such as the electricity market. Despite limiting the model proposed in Section III to 50 distinct, daily transactions per node, the network would still see approximately 1250–2500 transactions within a day. This translates to approximately 456,250–912,500 transactions within a year. The rapid growth in the network would then result in a slowdown in network transaction confirmations. In a real time trade market such as the microgrid, long transaction approval delays would be intolerable. A commodity such as electrical energy requires a transfer as soon as is physically possible to ensure good continuity in the supply. The expected growth would also affect the design of the smart microgrid in terms of the number of mining centres required to service the network at increasing scales and the processing capability and energy requirements that would be needed for each centre. As such, a more scalable DLT structure with smart contract functionality or a faster consensus mechanism would be required for a sustainable energy market solution. Consideration of the telecommunications infrastructure that microgrid transactions would require is also key for the successful realisation of the project. While 4G network connectivity is available, it is highly concentrated in the more urban provinces of the country. Most of the country is either being serviced using 3G and 2G connectivity or being without network connectivity [54]. As can be seen in Figure 10, the limited cell connectivity sometimes forces residents of a rural community to travel closer to the cell phone towers (if easily accessible) in order to be able to a signal strong enough to support and conduct any activities that requires a connection to the internet. Looking further into the mobile network connectivity available in the country, it can be seen that the download speed offered on the 3G networks peaks at 6.4 Mbps [37]. Microgrid communications would therefore need to be able to operate on this limited bandwidth while also being able to continue and recover should there be an interruption in the connection. In areas where no existing network connectivity exists, adequate infrastructure would need to be rolled out and implemented to enable participation in the community microgrid.

6.3. IIoT-DLT Compatibility

In order to try and better manage the scalability of the DLT network, alternative distributed ledger structures have been implemented in other DLT market players such as IoTA and NANO (formally Raiblocks). Over the course of this work, it was found that while these technologies do currently exist, they are currently unsuited for use in IIoT edge application spaces such as the smart microgrid.

Table 2 gives a snapshot of the system requirements for some of the alternative DLTs against the resources available on some popular platforms used in the Industrial IoT edge [49,55–59]. It can be seen that the systems requirements are far greater than what even the Raspberry Pi 3 can provide as a higher end IIoT edge device. IoTA has been successfully deployed onto the more powerful Asus Tinker Board; however, its specification is far more powerful than that available on most IIoT edge devices [60]. NANO have an available port for the Raspberry Pi; however, they do not currently have mechanisms to create a private testnet. Considering the system requirements, low power devices—enabled using the popular Cortex-M cores—and their real time operating systems are simply unable to support alternative DLT applications at this current time. Therefore, development work would need to be done in order to create appropriate ports of these technologies to enable them for use with Industrial IoT devices and operating systems.



(**a**) (**b**)

Figure 10. Limited cell reception forces community members to travel kilometres to be closer to cell towers (circled in red) in order to conduct online activities. (**a**) Supervisor searches for a stable cell signal in order to attend student's online MSc defense session; (**b**) Members of the community seek shelter from the summer sun after traveling to the cell tower in order to participate in online university activities.

Table 2. DLT installation requirements versus IIoT edge node resources.

	RAM	**Flash/ROM**
DLT Implementations		
Hyperledger Fabric	4 GB	Unspecified
NANO	2 GB	8 GB
IoTA	2–4 GB	60 GB
IIoT Edge Processors		
Cortex A53	1 GB	Variable
Cortex M0	8 kB	64 kB
Cortex M3	8 kB	128 kB
Cortex M4	128 kB	1024 kB
Cortex M7	512 kB	2048 kB

6.4. Security Concerns within the Smart Microgrid

As more and more IIoT devices are connected to the smart microgrid, it becomes increasingly important to ensure security from the physical layer up through to the application layer. Each smart microgrid IIoT device offers a new avenue for malicious attack and exploitation for the wider network. Thus, the industry needs to prioritise IIoT security solutions that are scalable and economic to maintain the concern of "do no harm" that is currently prevalent in operational technology standards while maintaining the real time deadlines seen in IIoT applications.

Each endpoint in the smart microgrid network—from smart meter controllers up to the demand forecasting centres—should have an appropriate level of security incorporated from the design stages of the system. To be able to identify the appropriate security level required for the application space, a detailed risk assessment would need to be conducted according to industry standards for electrical energy and the Industrial IoT. Guaranteeing security in the IIoT application spaces, such as the smart microgrid, however, is an ongoing challenge owing to the nature of the network and the devices within it. Compared to desktop PCs, IIoT networks are highly constrained in terms of their memory and the processing power available. These devices are also deployed using batteries as their source of power and these need to last for very long periods of time. As a result, a security solution for the IIoT cannot run so long that the batteries get depleted and need to be changed every few months as deployments can easily be well into the hundreds of thousands of devices. Security solution designs in the smart microgrid would therefore need to be guided by regulations and guidelines for IIoT networks, such as those developed by the Industrial Internet Consortium (IIC).

Although DLTs serve to provide a good number of the security features required for the secure transactive market—such as cryptography and non-repudiation—there still remains some concerns that need to be addressed. Most prominently is the issue of privacy. In the public blockchain, the ledger is open for all to view and interact with. While this allows for a simple, customer joining process, it would expose sensitive data to those that may not be permitted to view, such as customer payment information and organisational operations data.

Small Ethereum networks utilising Proof of Work consensus mechanisms have been found to be at risk of takeover by malicious actors utilising powerful mining machines. While the estimated network size of the microgrid at a national or regional level would be able to provide some measure of protection against such a takeover, networks in small rural communities would still be vulnerable to a Denial of Service attack. To ensure that only permitted actors are able to participate in the microgrid, a consortium implementing and regulating access control parameters may need to be implemented. Additionally, an alternative consensus mechanism could be employed such as proof of stake. In expanding this work towards improvements in the performance of DLTs in industrial application spaces, the authors aim to develop a new consensus mechanism, which is based on existing IIoT operations and measurements that are used as part of a stake that can be placed on blocks that are to be added into the ledger.

Security vulnerabilities could be introduced to the network owing to flaws in the smart contract code; thus, rigorous debugging would need to be conducted to ensure the correct contract implementation and patching policies would need to be developed as a guideline on how to address incorrectly configured contracts. Finally, as an emerging, disruptive technology, exploration on DLTs is ongoing with constant changes being made to implementations to make them more efficient. As they mature, there remains the risk that a flaw or attack could be exposed, which could compromise the security of the network, such as 51%, Race and Sybil attacks, gasless send, and third party wallet vulnerabilities.

7. Conclusions

With the move towards greener, more efficient power systems, the smart microgrid is emerging as the next iteration of the power grid. As a highly critical and regulated application space, adequate security is needed to protect grid operations from malicious actors. As a result, standards are emerging to which new services and technologies need to conform to in order to ensure the continued provision of the high levels of availability required by various power systems. As one of the technologies identified for the secure, transactive microgrid market, DLTs would also be subject to meeting the standard requirements for availability. To determine its suitability for operation at the IIoT edge, performance evaluation and stress tests were performed on a Raspberry Pi 3 acting as an Ethereum node. It was seen that while normal edge node operations were not impacted by the Ethereum processes, the node was not suited for near-real time or real time operations and that the mining process could disrupt the scalability of the network while driving up the end users costs. While there are a number of alternate DLTs, their current implementation requirements make them unsuitable for use at the IIoT edge.

For DLTs to be more suitable for IIoT applications, further work would need to be conducted in assessing architectural designs that could allow for the distribution of operations between the edge, fog and cloud. Part of this includes evaluating different IIoT edge platforms for compatibility with Ethereum along with possible improvements that could be made to the overall performance of DLT network activities implemented at the network edge. Efficiency could also be improved by creating and implementing consensus mechanisms that utilise existing IIoT processes instead of the computational burden introduced by the mathematical puzzle utilised by Proof of Work solutions. Improvements to DLT consensus would serve in reducing the implementation requirements of these technologies, making them more compatible with IIoT SoC architectures. The exploration on the implementation of the different ledger and permissibility structures seen in alternative DLTs could also be conducted as part of the effort towards improving the scalability and privacy of the DLT-IIoT smart microgrid network; however, prior work towards developing compatible IIoT installations, with reduced RAM and Flash requirements, would be required.

Author Contributions: L.P.I.L., G.P.H. and S.J.I. conceptualised the idea of this research; L.P.I.L. performed the experiments and data analysis; L.P.I.L. and G.P.H. wrote the paper; G.P.H., S.J.I. and H.S.V. provided supervision and reviewed the paper. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded in part by the Council for Scientific and Industrial Research under Projects 05400 054SN K9RTM11 and 05400 054SN KC8PASN and by City University of Hong Kong Project CityU 11202618 (7005053).

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

DLTs	Distributed Ledger Technologies
EV	Electric Vehicle
IIoT	Industrial Internet of Things
IoE	Internet of Energy
LEAGS	Local Energy Aggregators
MCU	Microcontroller Unit
NIST	National Institute of Standards and Technology
OS	Operating System

References

1. Crompton, R. Explained: Why Eskom Is In so Much Trouble. Available online: https://www.iol.co.za/news/opinion/explained-why-eskom-is-in-so-much-trouble-19238470 (accessed on 2 January 2021).
2. South African Market Insights. South Africa's Population Density Map. Available online: https://www.southafricanmi.com/population-density-map.html (accessed on 1 June 2020).
3. Eskom. Eskom Power Stations. Available online: https://greenworldwarriors.com/wp-content/uploads/2019/01/eskom-power-generation-cp.jpg (accessed on 1 June 2020).
4. Statistics South Africa. Census. 2011. Available online: https://www.statssa.gov.za/publications/P03014/P030142011.pdf (accessed on 1 June 2020).
5. Statistics South Africa. Community Survey. 2016. Available online: http://cs2016.statssa.gov.za/wp-content/uploads/2016/07/NT-30-06-2016-RELEASE-for-CS-2016-_Statistical-releas_1-July-2016.pdf (accessed on 1 June 2020).
6. Njini, F. South Africans Told to Prepare for a New Week of Rolling Blackouts. Available online: https://www.bloomberg.com/news/articles/2019-12-09/south-africans-braced-for-yet-another-week-of-rolling-blackouts (accessed on 1 June 2020).
7. Majola, G. Eskom's Three-Fold strategy to Recover Municipal Debt. Available online: https://www.iol.co.za/business-report/economy/eskoms-three-fold-strategy-to-recover-municipal-debt-50910592 (accessed on 2 January 2021).
8. Claasen, L. Municipalities Owe Eskom R31.5bn. Available online: https://www.moneyweb.co.za/news/south-africa/municipalities-owe-eskom-r31-5bn/ (accessed on 2 January 2021).
9. Reuters. Factbox: South Africa's Power Crisis—Overhauling State Utility Eskom. Available online: https://www.reuters.com/article/us-safrica-eskom-factbox-idUSKBN1X821H (accessed on 2 January 2021).
10. Wright, J.; Calitz, J. Brief Analysis of Variable Renewable Energy Contribution during Loadshedding (Q1 2019). Council of Scientific & Industrial Research Website. Available online: https://www.csir.co.za/sites/default/files/Documents/Renewable%20Energy_Q1_2019.pdf (accessed on 3 February 2021).

11. Eyewitness News. Eskom Takes It Up Two Notches and Announces Stage 6 Load Shedding. Available online: https://ewn.co.za/2019/12/09/eskom-takes-it-up-two-notches-and-announces-stage-6-load-shedding (accessed on 1 June 2020).
12. Eyewitness News. What Stage 6 (and 7, 8) Load Shedding Means for Gauteng. Available online: https://ewn.co.za/2019/12/10/what-stage-6-load-shedding-means (accessed on 1 June 2020).
13. 2018 South African Energy Prices Statistics. Public of South Africa Energy Department Website. 2008. Available online: http://www.energy.gov.za/files/media/explained/2018-South-African-Energy-Prices-Statistics.pdf (accessed on 8 March 2019).
14. Schedule of Standard Prices for ESKOM Tariffs 1 April 2019 to 31 March 2020 for Non-Local Authority Supplies, and 1 July 2019 to 30 June 2020 for Local Authority Supplies, 2019. ESKOM Company Wensite. Available online: https://www.eskom.co.za/CustomerCare/TariffsAndCharges/Documents/Complete%20Tariff%202019%20web1.pdf (accessed on 31 July 2020).
15. Mwangi, G. Winter Shack Fires on the Rise in Western Cape. Available online: https://www.iol.co.za/capeargus/news/winter-shack-fires-on-the-rise-in-western-cape-28555835 (accessed on 31 July 2020).
16. Pillitteri, V.Y.; Brewer, T.L. Guidelines for Smart Grid Cybersecurity. National Institute of Stardards and Technology Website. 2014. Available online: https://www.nist.gov/publications/guidelines-smart-grid-cybersecurity (accessed on 13 March 2019).
17. Ledwaba, L.P.I.; Hancke, G.P.; Isaac, S.J.; Venter, H.S. Developing a Secure, Smart Microgrid Energy Market using Distributed Ledger Technologies. In Proceedings of the 2019 IEEE 17th International Conference on Industrial Informatics (INDIN), Helsinki-Espoo, Finland, 22–25 July 2019; Volume 1, pp. 1725–1728.
18. Marzal, S.; Salas, R.; González-Medina, R.; Garcerá, G.; Figueres, E. Current challenges and future trends in the field of communication architectures for microgrids. *Renew. Sustain. Energy Rev.* **2018**, *82*, 3610–3622. [CrossRef]
19. Chen, K.C.; Yeh, P.C.; Hsieh, H.Y.; Chang, S.C. Communication infrastructure of smart grid. In Proceedings of the 4th International Symposium on Communications, Control and Signal Processing (ISCCSP), Limassol, Cyprus, 3–5 March 2010; pp. 1–5. [CrossRef]
20. D'Oriano, L.; Mastandrea, G.; Rana, G.; Raveduto, G.; Croce, V.; Verber, M.; Bertoncini, M. Decentralized blockchain flexibility system for Smart Grids: Requirements engineering and use cases. In Proceedings of the International IEEE Conference and Workshop in Óbuda on Electrical and Power Engineering (CANDO-EPE), Budapest, Hungary, 20–21 November 2018; pp. 39–44. [CrossRef]
21. Jogunola, O.; Ikpehai, A.; Anoh, K.; Adebisi, B.; Hammoudeh, M.; Son, S.Y.; Harris, G. State-Of-The-Art and Prospects for Peer-To-Peer Transaction-Based Energy System. *Energies* **2017**, *10*, 2106. [CrossRef]
22. Greer, C.; Wollman, D.A.; Prochaska, D.E.; Boynton, P.A.; Mazer, J.A.; Nguyen, C.T.; FitzPatrick, G.J.; Nelson, T.L.; Koepke, G.H., Jr.; Bushby, S.T.; et al. NIST Framework and Roadmap for Smart Grid Interoperability Standards, Release 3.0, 2014. National Institute of Stardards and Technology Website. Available online: https://www.nist.gov/publications/nist-framework-and-roadmap-smart-grid-interoperability-standards-release-30 (accessed on 13 March 2019).
23. Health, C. Why Blockchains Don't Suck, and the Perils of Distributed Databases. Available online: https://link.medium.com/LMDfudPUx5 (accessed on 6 November 2018).
24. Zhou, Z.; Tan, L.; Xu, G. Blockchain and Edge Computing Based Vehicle-to-Grid Energy Trading in Energy Internet. In Proceedings of the 2nd IEEE Conference on Energy Internet and Energy System Integration (EI2), Beijing, China, 20–22 October 2018; pp. 1–5. [CrossRef]
25. Wang, Y.; Su, Z.; Xu, Q.; Zhang, N. Contract Based Energy Blockchain for Secure Electric Vehicles Charging in Smart Community. In Proceedings of the IEEE 16th International Conference on Dependable, Autonomic and Secure Computing, 16th International Conference on Pervasive Intelligence and Computing, 4th International Conference on Big Data Intelligence and Computing and Cyber Science and Technology Congress (DASC/PiCom/DataCom/CyberSciTech), Athens, Greece, 12–15 August 2018; pp. 323–327. [CrossRef]
26. Yu, R.; Zhong, W.; Xie, S.; Yuen, C.; Gjessing, S.; Zhang, Y. Balancing Power Demand Through EV Mobility in Vehicle-to-Grid Mobile Energy Networks. *IEEE Trans. Ind. Inform.* **2016**, *12*, 79–90. [CrossRef]
27. Kang, E.S.; Pee, S.J.; Song, J.G.; Jang, J.W. A Blockchain-Based Energy Trading Platform for Smart Homes in a Microgrid. In Proceedings of the 3rd International Conference on Computer and Communication Systems (ICCCS), Nagoya, Japan, 27–30 April 2018; pp. 472–476. [CrossRef]
28. Zhou, L.; Yeh, K.H.; Hancke, G.; Liu, Z.; Su, C. Security and Privacy for the Industrial Internet of Things: An Overview of Approaches to Safeguarding Endpoints. *IEEE Signal Process. Mag.* **2018**, *35*, 76–87. [CrossRef]
29. Cheng, B.; Zhang, J.; Hancke, G.P.; Karnouskos, S.; Colombo, A.W. Industrial Cyberphysical Systems: Realizing Cloud-Based Big Data Infrastructures. *IEEE Ind. Electron. Mag.* **2018**, *12*, 25–35. [CrossRef]
30. Hu, Q.; Zhang, J.; Mitrokotsa, A.; Hancke, G. Tangible security: Survey of methods supporting secure ad-hoc connects of edge devices with physical context. *Comput. Secur.* **2018**, *78*, 281–300. [CrossRef]
31. Zhou, L.; Su, C.; Li, Z.; Liu, Z.; Hancke, G.P. Automatic fine-grained access control in SCADA by machine learning. *Future Gener. Comput. Syst.* **2019**, *93*, 548–559. [CrossRef]
32. Abu-Mahfouz, A.M.; Hancke, G.P. ns-2 extension to simulate localization system in wireless sensor networks. In Proceedings of the IEEE Africon '11, Victoria Falls, Zambia, 13–15 September 2011; pp. 1–7. [CrossRef]
33. Hancke, G.P.; Silva, B.J. Wireless Positioning in Underground Mines: Challenges and Recent Advances. *IEEE Ind. Electron. Mag.* **2021**. [CrossRef]
34. Li, Z.; Kang, J.; Yu, R.; Ye, D.; Deng, Q.; Zhang, Y. Consortium Blockchain for Secure Energy Trading in Industrial Internet of Things. *IEEE Trans. Ind. Inform.* **2018**, *14*, 3690–3700. [CrossRef]

35. Sankaran, S.; Sanju, S.; Achuthan, K. Towards Realistic Energy Profiling of Blockchains for Securing Internet of Things. In Proceedings of the IEEE 38th International Conference on Distributed Computing Systems (ICDCS), Vienna, Austria, 2–5 July 2018; pp. 1454–1459. [CrossRef]
36. Isaac, S.J. City University of Hong Kong, Hong Kong, China. Smart Transactive Microgrid with Stability Control for Decentralised Local Peer-to-Peer Markets with Dynamic Pricing. Unpublished work, 2020.
37. OpenSignal. State of Mobile Networks: South Africa (August 2017). Available online: https://www.opensignal.com/reports/2017/08/southafrica/state-of-the-mobile-network (accessed on 1 June 2020).
38. Wahab, A.; Mehmood, W. Survey of consensus protocols. *arXiv* **2018**, arXiv:1810.03357.
39. Ray, J. Consortium Chain Development. Available online: https://github.com/ethereum/wiki/wiki/Consortium-Chain-Development#efficiency-improvements (accessed on 1 June 2020).
40. Alex Chumbley, K.M.; Khim, J. Merkle Tree. Available online: https://brilliant.org/wiki/merkle-tree/ (accessed on 1 June 2020).
41. Station, E.G. How Long does an Ethereum Transaction Really Take? Available online: https://ethgasstation.info/blog/ethereum-transaction-how-long/ (accessed on 1 June 2020).
42. Etherstats. Network Statistics. Available online: https://ethstats.io/?utm_source=content&utm_medium=medium (accessed on 1 June 2020).
43. Siriwardena, P. The Mystery Behind Block Time. Available online: https://medium.facilelogin.com/the-mystery-behind-block-time-63351e35603a (accessed on 1 June 2020).
44. Palau, A. Analyzing the Hardware Requirements to Be an Ethereum Full Validated Node II. Available online: https://link.medium.com/BdMobeLUx5 (accessed on 13 March 2019).
45. Etherscan. Blocks. Available online: https://etherscan.io/blocks (accessed on 1 June 2020).
46. Mitra, R. Bitcoin VS Ethereum: [The Ultimate Step-by-Step Comparison Guide]. Available online: https://blockgeeks.com/guides/bitcoin-vs-ethereum-ultimate-comparison-guide/ (accessed on 13 March 2019).
47. Kraken. Cryptocurrency Deposit Processing Times. Available online: https://support.kraken.com/hc/en-us/articles/203325283-Cryptocurrency-deposit-processing-times (accessed on 2 June 2020).
48. Charts, B.I. Ethereum Block Size Historical Chart. Available online: https://bitinfocharts.com/comparison/size-eth-sma7.html (accessed on 2 June 2020).
49. Ledwaba, L.P.I.; Hancke, G.P.; Venter, H.S.; Isaac, S.J. Performance costs of software cryptography in securing new-generation Internet of energy endpoint devices. *IEEE Access* **2018**, *6*, 9303–9323. [CrossRef]
50. Eloudrhiri, S. Create a Private Ethereum Blockchain with IoT Devices (1/6). Available online: https://chainskills.com/2017/02/24/create-a-private-ethereum-blockchain-with-iot-devices-16/ (accessed on 13 March 2019).
51. Patel, K. Monitor the Core Temperature of Your Raspberry Pi. Available online: https://medium.com/@kevalpatel2106/monitor-the-core-temperature-of-your-raspberry-pi-3ddfdf82989f (accessed on 14 March 2019).
52. Ņikiforovs, P. Htop Explained. Available online: https://peteris.rocks/blog/htop/#mem-memory-usage (accessed on 3 February 2021).
53. Wood-Craig, N. Pi-Archimedes. Available online: http://www.craig-wood.com/articles/pi-archimedes/index.html (accessed on 7 March 2019).
54. nPerf. 2G/3G/4G Coverage Map, South Africa. Available online: https://www.nperf.com/en/map/ZA/-/5591.Vodacom/signal/?ll=-28.30214755175344&lg=12.229676909900311&zoom=5 (accessed on 1 June 2020).
55. Hyperledger Foundation. Installing Pre-Requisites | Hyperledger. Available online: https://hyperledger.github.io/composer/v0.19/installing/installing-prereqs (accessed on 11 March 2019).
56. IoTA Foundation. IoTA FAQs. Available online: https://www.iota.org/get-started/faqs (accessed on 8 March 2019).
57. IoTA. IOTA Full Node Installation Wiki—Python documentation. Available online: https://iri-playbook.readthedocs.io/en/master/ (accessed on 6 March 2019).
58. Nano Developers. Node Setup. Available online: https://developers.nano.org/guides/node-setup/ (accessed on 11 March 2019).
59. NANO. Nano | FAQ. Available online: https://nano.org/en/faq (accessed on 11 March 2019).
60. yillkid. Deploy IOTA Full Node to Cheap ARM boards. Available online: https://medium.com/biilabs/deploy-iota-fullnode-on-asus-tinker-board-fcd2cff8331f (accessed on 13 March 2019).