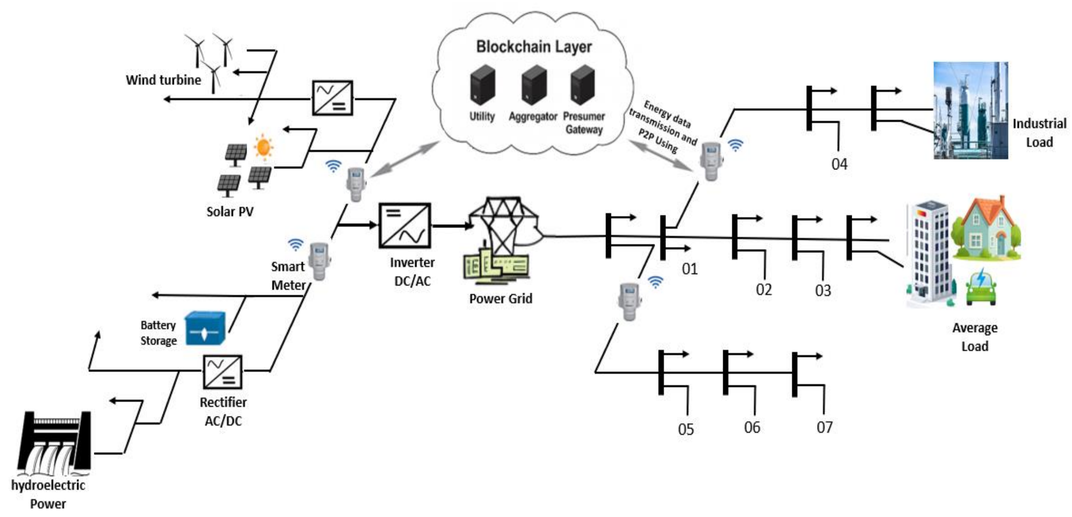
seanergy
19›20 May, 2026
Nantes • France

Poster n°17

Enhancing the Reliability of Decentralized Hybrid Renewable Energy Metering Systems in Microgrids through Blockchain and Smart Meter Technologies

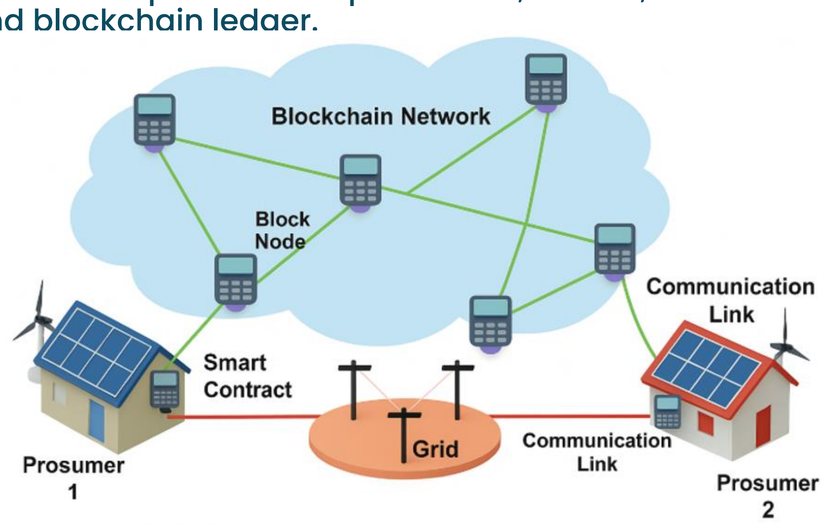
Context and abstract

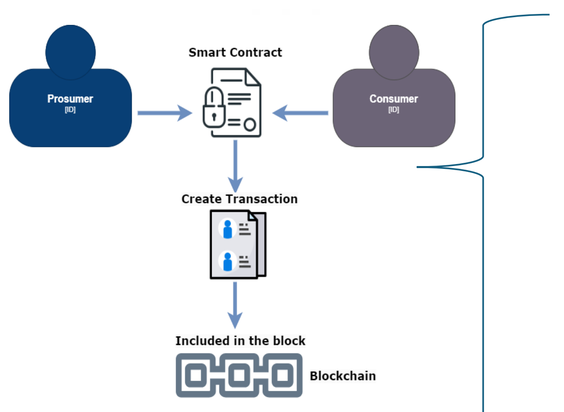
This work proposes a blockchain-based smart microgrid integrating renewable sources (solar, wind, hydro), smart meters, and peer-to-peer energy trading. Using a Proof-of-Authority consensus, the system enables fast, secure, and low-cost energy transactions with real-time monitoring. Simulation results show 82.4% renewable penetration, 99.7% billing accuracy, and a 17% cost reduction for users. Compared to centralized systems, it improves efficiency, reduces latency, and enhances transparency for future decentralized energy management.



Methods

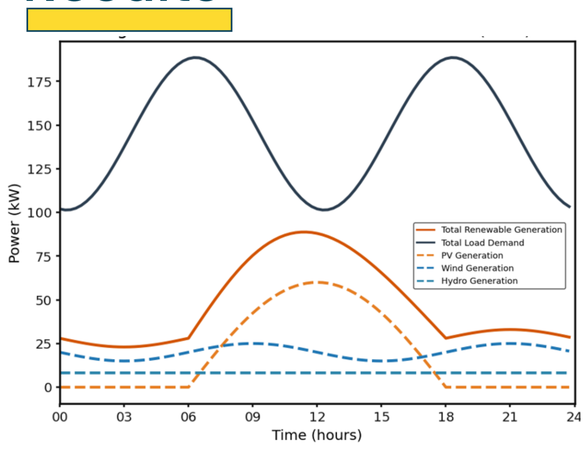
The proposed hybrid microgrid utilizes blockchain technology and incorporates photovoltaic panels, wind turbines, a hydroelectric power plant, and a battery storage unit. Energy management loop between production, meter, consumer loads, and blockchain ledaer.



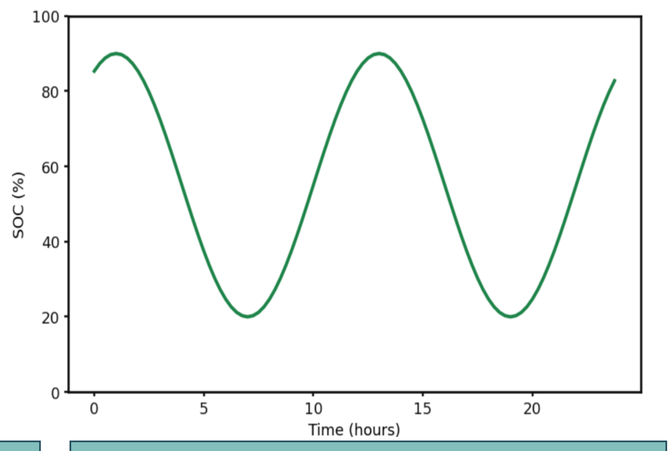



```
#main.py
import hashlib                                          # This class represents a block in the blockchain

class HydrogenBlock:
    def __init__(self, previous_block_hash, transaction_list):          # initializes the attributes of the block
        self.previous_block_hash = previous_block_hash                  # to store the hash of the previous block
        self.transaction_list = transaction_list                        # to store the transactions of the block
        self.block_data = f" {' - '.join(transaction_list)} - {previous_block_hash}" # to store the block data as a string
        self.block_hash = hashlib.sha256(self.block_data.encode()).hexdigest()   # to store the hash of the current block
class Blockchain:
        def __init__(self):                             # initializes the string by creating the genesis block using the method
        self.chain = []
        self.generate_hydrogen_block()
    def generate_hydrogen_block(self):  # creates the genesis block with a special transaction ('Genesis Block') and adds it to the chain

        self.chain.append(HydrogenBlock("0", ['H2 Transaction Block']))
    def create_block_from_transaction(self, transaction_list):
        previous_block_hash = self.last_block.block_hash
        self.chain.append(HydrogenBlock(previous_block_hash, transaction_list))  # creates a new block from a list of transactions and
I add to the chain using the hash of the last block as the previous hash.
    def display_chain(self):                                    # displays the data and hashes of all the blocks in the chain
        for i in range(len(self.chain)):
            print(f"Data {i + 1}: {self.chain[i].block_data}")
            print(f"Hash {i + 1}: {self.chain[i].block_hash}\n")
    @property
    def last_block(self):
        return self.chain[-1]
t1 = "prosum_1 send 100 Kg of hydrogen to consum_1"
t2 = "prosum_1 send 90 Kg of hydrogen to consum_2"
t3 = "prosum_1 send 70 Kg of hydrogen to consum_3"
myblockchain = Blockchain()                                     # create a new instance of the Blockchain class
myblockchain.create_block_from_transaction([t1])    # Add two new blocks to the chain, each containing a transaction
myblockchain.create_block_from_transaction([t2])
myblockchain.create_block_from_transaction([t3])
myblockchain.display_chain()                                    # displays the data and hashes of all the blocks in the chain
```

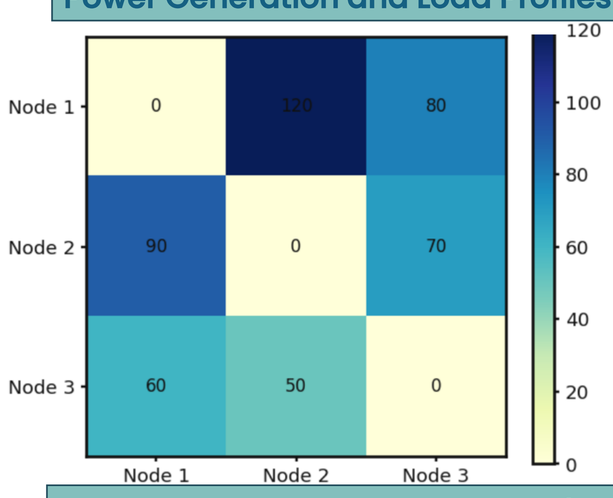
Results



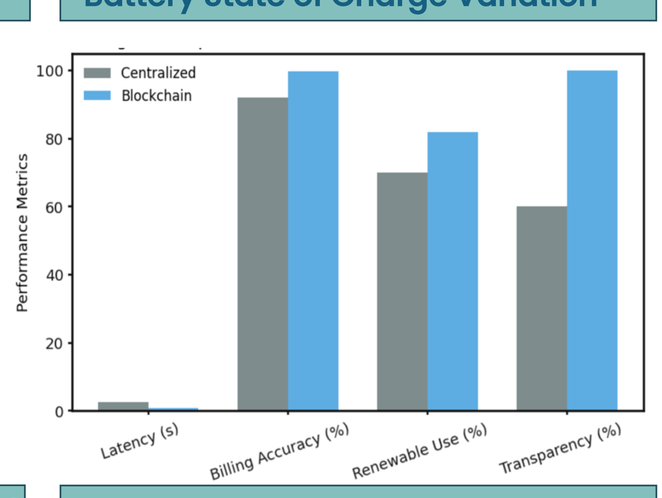
Power Generation and Load Profiles



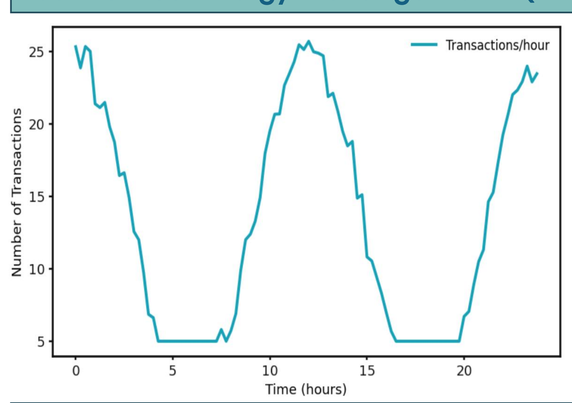
Battery State of Charge Variation



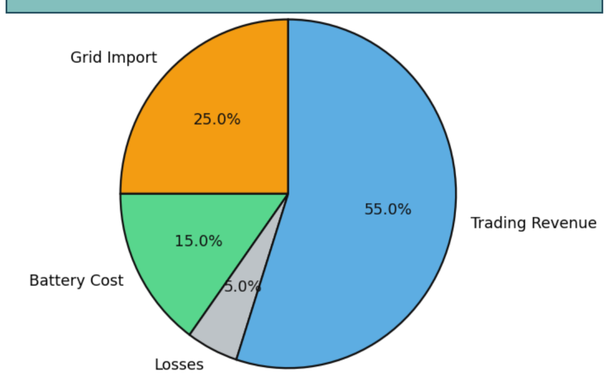
Peer-to-Peer Energy Exchange Matrix (kWh)



Centralized vs Blockchain-Based Control



Blockchain Transaction Activity over Time



Cost and Revenue Distribution.

Conclusion

- ✓ Achieves 82.4% renewable penetration with 99.7% billing accuracy.
- ✓ Enables 1,152 transactions/day with 0.83 s latency using PoA.
- ✓ Reduces energy costs by 17% and latency by 69%.
- ✓ Supports scalable and efficient decentralized microgrids with 88.5% battery efficiency.

Future work

Future work will focus on testing the system in real industrial and community microgrids. It will also integrate IoT protocols like MQTT and DLMS/COSEM to improve communication. Hybrid consensus (PoA–PoS) and edge computing will be used to enhance scalability and security. Finally, the system will be extended to include EV charging and green hydrogen for net-zero energy systems.