MANTECH Publications

Journal of Blockchain Systems and Smart Contracts
Volume 2, Issue 2, May-August 2026
ISSN: 3139-132X (Online)

Energy-Efficient Blockchain Consensus Mechanisms for Scalable Smart Contract Execution in IoT Networks

Dr. Aarav Mehta[1], Nisha Verma[2], Rohan Kulshreshtha[3], Kavita Rao[4]

ABSTRACT

Internet of Things (IoT) deployments generate large volumes of sensor and actuator events, while many devices operate under strict energy, memory, bandwidth, and latency constraints. Blockchain-based systems can support tamper-evident coordination and smart-contract automation across organizations, but consensus and replicated execution can add communication, computation, storage, and energy overhead. This paper reviews the design trade-offs of common consensus families—including proof of work (PoW), proof of stake (PoS), proof of authority (PoA), practical Byzantine fault-tolerant (PBFT-style) protocols, crash-fault-tolerant leader-based protocols, and directed acyclic graph (DAG)-oriented approaches—in IoT contexts. It proposes a reference architecture in which constrained devices perform sensing and lightweight validation, edge gateways aggregate and batch events, and appropriately selected validator nodes order transactions and execute smart contracts. The paper defines a reproducible evaluation plan using energy per confirmed transaction, throughput, end-to-end latency, communication overhead, fault tolerance, and contract execution cost. Because no experimental measurements were supplied or collected for this manuscript, comparative descriptions are qualitative and no numerical performance advantage is claimed. The central conclusion is that consensus selection should be driven by the trust model and workload rather than by a single universal ranking.

KEYWORDS: *Blockchain; Internet of Things; consensus mechanisms; energy efficiency; smart contracts; edge computing; scalability; PBFT; proof of authority.*

Page 57-69 © MANTECH PUBLICATIONS 2026. All Rights Reserved

INTRODUCTION

IoT systems connect sensors, meters, vehicles, industrial controllers, medical devices, and smart-building infrastructure. Many deployments need a reliable way to coordinate events across administrative boundaries, preserve audit trails, and automate conditional actions. Smart contracts can encode business rules such as equipment maintenance triggers, energy-market settlement, supply-chain status transitions, or access-control decisions. Blockchain offers a replicated log and shared execution environment, but its costs can be unsuitable for constrained devices if every node is expected to store data, validate all transactions, and participate in consensus.

The term energy-efficient blockchain should be interpreted carefully. Low consensus energy does not automatically imply low total system energy: gateway hardware, network traffic, cryptographic verification, data replication, cooling, and storage also consume resources. Similarly, high transactions per second may conceal longer confirmation times, weaker fault assumptions, or increased validator centralization. IoT deployments therefore require multi-dimensional evaluation under an explicit adversary and trust model.

This paper provides a comparative review and proposed architecture for energy-aware smart-contract execution in IoT networks. It emphasizes the distinction between permissionless public chains and permissioned or consortium deployments, where validator identity and governance assumptions differ.

PROBLEM STATEMENT

The design challenge is to execute smart contracts reliably at IoT scale without imposing excessive energy and communication demands on constrained endpoints. PoW is generally unsuitable for direct participation by battery-powered devices; permissioned protocols may reduce energy cost but depend on validator governance and membership controls; BFT protocols may incur communication overhead as validator sets grow; and edge aggregation can reduce on-chain traffic while creating gateway trust and availability concerns. A scalable design must balance energy consumption, latency, throughput, decentralization, resilience, privacy, and operational complexity.

Page 57-69 © MANTECH PUBLICATIONS 2026. All Rights Reserved

AIM AND OBJECTIVES

Aim: To assess consensus design trade-offs and propose an energy-aware architecture for scalable smart-contract execution in IoT networks.

- Compare consensus families by energy profile, fault assumptions, communication overhead, and suitability for IoT workloads.
- Describe an architecture that separates constrained sensing devices from consensus and contract-execution infrastructure.
- Identify bottlenecks related to batching, storage growth, network partitioning, and replicated execution.
- Define a reproducible evaluation methodology and metrics for future experimental validation.
- Discuss security and governance risks associated with edge gateways, validator selection, and permissioned membership.

METHODOLOGY

This paper uses a focused narrative review of established blockchain consensus literature, foundational consensus protocols, and standards-oriented IoT and edge-computing concepts. It is not a systematic review and does not claim exhaustive coverage. The proposed architecture and evaluation plan are design recommendations synthesized from the reviewed principles.

No physical IoT testbed, energy meter readings, blockchain deployment logs, or experimental dataset were available for this manuscript. Therefore, no measured energy values, throughput figures, p-values, or fabricated benchmark outcomes are reported. The consensus comparison below is qualitative and should be validated under a common workload and hardware configuration.

Page 57-69 © MANTECH PUBLICATIONS 2026. All Rights Reserved

CONSENSUS MECHANISMS: DESIGN TRADE-OFFS

Table: 1

Consensus family	Energy / compute profile	Strengths	Key limitations in IoT settings
Proof of Work (PoW)	Typically high aggregate computation and electricity demand	Open participation; well-studied security model in public networks	Unsuitable for low-power endpoints; confirmation delay and resource overhead
Proof of Stake (PoS)	Avoids PoW-style continuous hashing; validator work still requires computation and networking	Can support public participation and economic security	Stake concentration, protocol complexity, validator hardware/network demands
Proof of Authority (PoA)	Usually low energy because a known validator set produces blocks	Low overhead and predictable finality in consortium deployments	Trust and governance shift to validator selection and accountability
PBFT-style Byzantine fault tolerance	No mining; message complexity can grow rapidly with validator count	Deterministic finality under stated Byzantine-fault assumptions	Coordination and network overhead; sensitive to latency and membership size
Leader-based crash-fault-tolerant (e.g., Raft)	Generally modest energy and communication cost	Simple operation for trusted clusters; efficient under crash failures	Does not tolerate arbitrary Byzantine behavior without additional controls
DAG / gossip-oriented approaches	Varies by protocol; may improve parallelism or data dissemination	Potential concurrency and flexible event propagation	Finality and security guarantees vary; implementation complexity and metadata overhead

Page 57-69 © MANTECH PUBLICATIONS 2026. All Rights Reserved

PROPOSED ENERGY-AWARE IOT BLOCKCHAIN ARCHITECTURE

The proposed architecture uses a tiered design. Sensors and actuators sign or authenticate measurements and transmit compact events. Edge gateways validate schemas, remove duplicates, aggregate readings, and batch transactions according to latency and risk requirements. A blockchain validator layer establishes ordering and finality using a consensus protocol selected for the deployment's trust model. Smart contracts execute business rules and emit auditable events, while monitoring components track energy, latency, failures, and resource utilization.

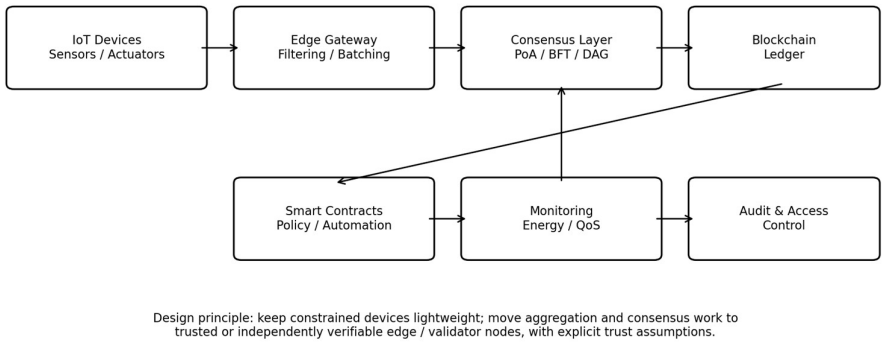


Figure 1: Reference architecture for energy-aware IoT blockchain deployments

Endpoint And Edge Responsibilities

Battery-powered endpoints should not be required to perform continuous consensus work unless the device capability and threat model justify it. Endpoints can use authenticated messaging, compact signatures, timestamps or sequence numbers, and local buffering. Gateways can batch low-risk telemetry while forwarding urgent alarms with lower delay. The gateway must not silently rewrite measurements: signed payloads, hash commitments, secure boot, key management, and independent audit trails can reduce but not eliminate gateway compromise risks.

Consensus Selection

For a consortium with known organizations and an accountable validator set, PoA or a BFT protocol may offer a reasonable energy profile, provided the trust assumptions are acceptable. If the system must resist arbitrary validator behavior, a Byzantine-fault-tolerant protocol is more appropriate than a crash-fault-only protocol. For a public, permissionless network, the application may be better served by anchoring periodic commitments to an established chain

Page 57-69 © MANTECH PUBLICATIONS 2026. All Rights Reserved

rather than attempting to operate a custom low-energy consensus layer without a credible security budget.

Smart-Contract Execution

Smart contracts should be small, bounded, and explicit about failure handling. They should validate input ranges, reject stale or duplicate events, separate critical control logic from analytics, and emit events needed for audit. Expensive analytics are often better performed off-chain with verifiable summaries or proofs, while the contract stores the minimum state required for settlement, access control, or accountability. Gas or execution limits should be set to prevent a single transaction from exhausting block resources.

TRANSACTION LIFECYCLE AND SCALING

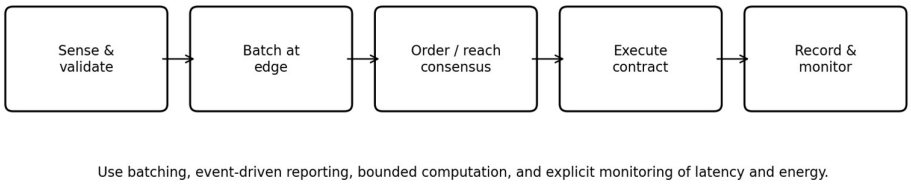


Figure 2: Energy-aware transaction lifecycle

Table: 2

Stage	Energy-aware design choice	Risk / safeguard
Sensing	Local filtering, calibration, compact signed payloads	Do not discard safety-critical events solely to save bandwidth
Edge aggregation	Batch routine telemetry and compress repeated values	Preserve raw evidence or verifiable hashes where required
Consensus	Choose protocol according to trust and fault assumptions	Avoid claiming equivalent security across different consensus families
Contract execution	Bound computation; keep analytics off-chain where feasible	Validate data provenance and prevent replay

Page 57-69 © MANTECH PUBLICATIONS 2026. All Rights Reserved

Storage	Store compact state and periodic commitments; define retention	Off-chain storage requires availability and integrity mechanisms
Monitoring	Measure energy, latency, retries, queue length, and failures	Report tail latency and failure cases, not averages alone

ENERGY AND SCALABILITY EVALUATION FRAMEWORK

A valid comparison should hold hardware, network topology, workload, contract logic, and security assumptions constant where possible. Test scenarios should vary device count, transaction rate, validator count, message loss, packet delay, node failure, and bursty sensor traffic. Energy should be measured at the device and infrastructure levels separately, because a gateway-heavy architecture may reduce endpoint energy while increasing total system consumption.

Table: 3

Metric	Definition / measurement approach	Reporting guidance
Energy per confirmed transaction	Joules consumed by specified components divided by confirmed transactions	State measurement boundary and include idle baseline
Throughput	Successfully finalized transactions per second	Distinguish submitted, accepted, executed, and finalized transactions
End-to-end latency	Time from event creation to contract outcome or finality	Report median, 95th, and 99th percentiles
Communication overhead	Bytes transmitted and messages exchanged per transaction	Include retries, gossip, and protocol metadata
Resource utilization	CPU, RAM, storage I/O, and network use	Measure endpoint, gateway, and validator separately
Fault tolerance	Behavior under crashes, Byzantine nodes, delay, and partitions	Specify fault assumptions and safety/liveness outcomes

Page 57-69 © MANTECH PUBLICATIONS 2026. All Rights Reserved

Execution cost	Contract computation or gas, state growth, and failed execution rate	Use identical contract workload for comparisons
Scalability	Performance as endpoints, validators, and event rates increase	Identify the first bottleneck and tail-latency degradation

Experimental Design

A future study can implement representative workloads such as smart-meter reporting, industrial anomaly alerts, and access-control events. Candidate systems should include at least one permissioned PoA configuration, one BFT configuration, and a baseline public-chain anchoring approach. Each run should be repeated under a documented configuration, with confidence intervals or dispersion statistics reported. The protocol version, hardware, software configuration, validator count, event payload size, and measurement instrumentation should be published to support reproducibility.

Hypotheses For Future Testing

- H1: Batching routine telemetry at the edge will reduce on-chain transaction volume compared with submitting every sensor sample individually.
- H2: A permissioned low-energy consensus configuration will reduce consensus-related energy compared with a PoW baseline under comparable workloads, while offering different decentralization and trust guarantees.
- H3: Increasing validator count in a BFT protocol will increase communication overhead, with the magnitude dependent on protocol implementation and network conditions.
- H4: Batching thresholds that reduce energy may increase latency for individual events, requiring workload-specific tuning.

These are proposed hypotheses, not findings. They should be tested experimentally rather than stated as confirmed results.

Page 57-69 © MANTECH PUBLICATIONS 2026. All Rights Reserved

SECURITY, PRIVACY, AND RELIABILITY

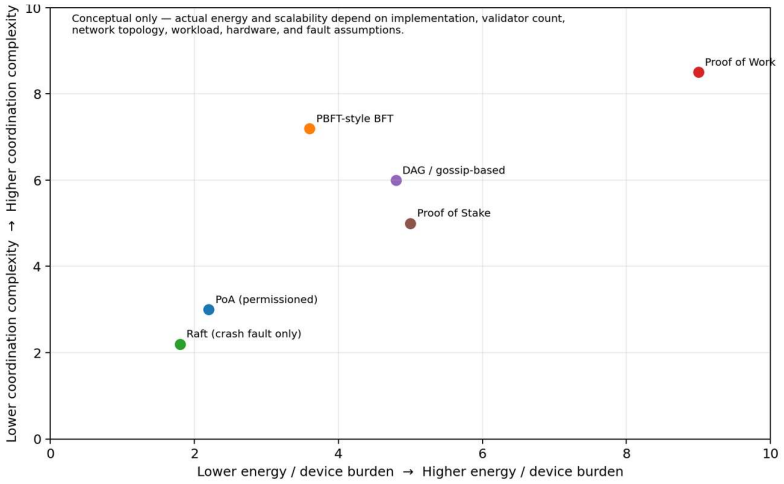


Figure 3: Qualitative consensus design space. Positions are conceptual, not measured benchmark scores

Table: 4

Risk	Potential consequence	Mitigation options
Compromised gateway	False or suppressed sensor events	Device signatures, secure boot, redundant sensing, anomaly detection, audit commitments
Validator collusion	Censorship or invalid state transitions within protocol limits	Independent organizations, threshold policies, rotation, monitoring, incident response
Network partition	Delayed finality, stale data, split operational view	Explicit timeout policy, local safe mode, replay protection, recovery testing
Contract vulnerability	Unauthorized execution or asset loss	Code review, automated testing, formal verification for critical logic, upgrade timelocks
Key compromise	Forged device events or administrator actions	Hardware-backed keys where feasible, rotation, revocation, least privilege

Page 57-69 © MANTECH PUBLICATIONS 2026. All Rights Reserved

Privacy exposure	Sensitive operational or personal data becomes linkable	Data minimization, access controls, encryption off-chain, only anchor necessary commitments
Denial of service	Queue growth and urgent-event delay	Rate limits, priority lanes for critical events, bounded contract execution, load shedding policies

Consensus efficiency must not be evaluated separately from safety. In industrial or infrastructure settings, a blockchain outage should not force an actuator into an unsafe state. Critical local control should remain available through a documented fail-safe mode, and blockchain confirmation should not be the sole control loop for time-critical physical processes.

DISCUSSION

The most suitable consensus mechanism depends on who is allowed to validate, what fraction of validators may be malicious or unavailable, whether finality must be deterministic, and how much delay the application can tolerate. Permissioned protocols can be energy-efficient, but the governance and operational controls that determine validator membership become part of the security boundary. Public networks may provide broader independent verification but can impose different transaction costs and latency constraints.

A tiered architecture offers a practical way to reduce the work imposed on constrained endpoints. Yet batching and edge processing are not free: they can introduce latency, buffering risks, gateway trust dependencies, and loss of fine-grained auditability if evidence is discarded. The design should therefore separate routine telemetry from urgent alerts and define data-retention and audit requirements before optimizing energy.

The paper’s central recommendation is to report energy together with security assumptions and service quality. A mechanism that consumes less energy but fails under the required adversary model is not a valid optimization. Similarly, a throughput improvement is incomplete if tail latency, storage growth, or total infrastructure energy becomes unacceptable.

Page 57-69 © MANTECH PUBLICATIONS 2026. All Rights Reserved

LIMITATIONS

This manuscript is a narrative review and proposed evaluation framework rather than an experimental study. It contains no measured benchmark results and does not establish a universal ranking of consensus protocols. Performance depends on protocol version, hardware, network conditions, validator count, contract workload, and measurement boundaries. The author details are fictional placeholders for document-formatting purposes and must be replaced before any academic submission.

FUTURE SCOPE

- Build a reproducible testbed with low-power IoT devices, edge gateways, and multiple validator configurations.
- Measure endpoint, gateway, and validator energy separately under steady and burst workloads.
- Test fault injection, compromised gateways, malicious validators, network partitions, and replay attacks.
- Compare direct on-chain telemetry, edge batching, and periodic commitment anchoring.
- Evaluate privacy-preserving data sharing and verifiable off-chain computation for sensitive IoT applications.
- Publish code, configuration files, data, and measurement methods, subject to security and privacy constraints.

CONCLUSION

Energy-efficient blockchain execution for IoT requires a system-level design rather than a consensus choice in isolation. A practical architecture keeps constrained endpoints lightweight, uses edge aggregation carefully, selects consensus according to explicit trust and fault assumptions, bounds smart-contract computation, and measures energy alongside latency, throughput, communication overhead, and resilience. Permissioned PoA and BFT-style protocols can be attractive for consortium deployments, while public-chain anchoring may suit systems that need independent public auditability without operating a new consensus network. These choices involve different trade-offs and must be validated in the intended environment. The framework and hypotheses presented here provide a basis for reproducible experimental evaluation; they should not be mistaken for measured findings.

Page 57-69 © MANTECH PUBLICATIONS 2026. All Rights Reserved

REFERENCES

1. Nakamoto S. Bitcoin: A Peer-to-Peer Electronic Cash System. 2008.
2. Castro M, Liskov B. Practical Byzantine Fault Tolerance. Proceedings of the Third Symposium on Operating Systems Design and Implementation (OSDI). 1999:173–186.
3. Ongaro D, Ousterhout J. In Search of an Understandable Consensus Algorithm. Proceedings of USENIX Annual Technical Conference. 2014:305–319.
4. Cachin C, Vukolić M. Blockchain Consensus Protocols in the Wild. arXiv:1707.01873. 2017.
5. Christidis K, Devetsikiotis M. Blockchains and Smart Contracts for the Internet of Things. IEEE Access. 2016;4:2292–2303. doi:10.1109/ACCESS.2016.2566339.
6. Dorri A, Kanhere SS, Jurdak R. Blockchain in Internet of Things: Challenges and Solutions. arXiv:1608.05187. 2016.
7. Atzori L, Iera A, Morabito G. The Internet of Things: A Survey. Computer Networks. 2010;54(15):2787–2805. doi:10.1016/j.comnet.2010.05.010.
8. Yaga D, Mell P, Roby N, Scarfone K. Blockchain Technology Overview. NISTIR 8202. National Institute of Standards and Technology. 2018. doi:10.6028/NIST.IR.8202.
9. Wood G. Ethereum: A Secure Decentralised Generalised Transaction Ledger. Ethereum Yellow Paper. Technical specification; consult current version for implementation details.
10. International Organization for Standardization. ISO/IEC 30141: Internet of Things (IoT)—Reference Architecture. 2018.

Page 57-69 © MANTECH PUBLICATIONS 2026. All Rights Reserved

***Author for Correspondence**

Dr. Aarav Mehta

E-mail: aarav.mehta2sept@gmail.com

[1]Associate Professor, Department of Computer Science, North Valley Institute of Technology, Jaipur, Rajasthan, India

[2]Research Scholar, Department of Computer Science, North Valley Institute of Technology, Jaipur, Rajasthan, India

[3]Research Scholar, Department of Computer Science, North Valley Institute of Technology, Jaipur, Rajasthan, India

[4]Research Scholar, Department of Computer Science, North Valley Institute of Technology, Jaipur, Rajasthan, India

Received Date: June 10, 2026

Accepted Date: June 22, 2026

Published Date: July 03, 2026

Citation: Dr. Aarav Mehta, Nisha Verma, Rohan Kulshreshtha, Kavita Rao. Energy-Efficient Blockchain Consensus Mechanisms for Scalable Smart Contract Execution in IoT Networks. Journal of Blockchain Systems and Smart Contracts. 2026; 2(2): 57-69p.

Page 57-69 © MANTECH PUBLICATIONS 2026. All Rights Reserved