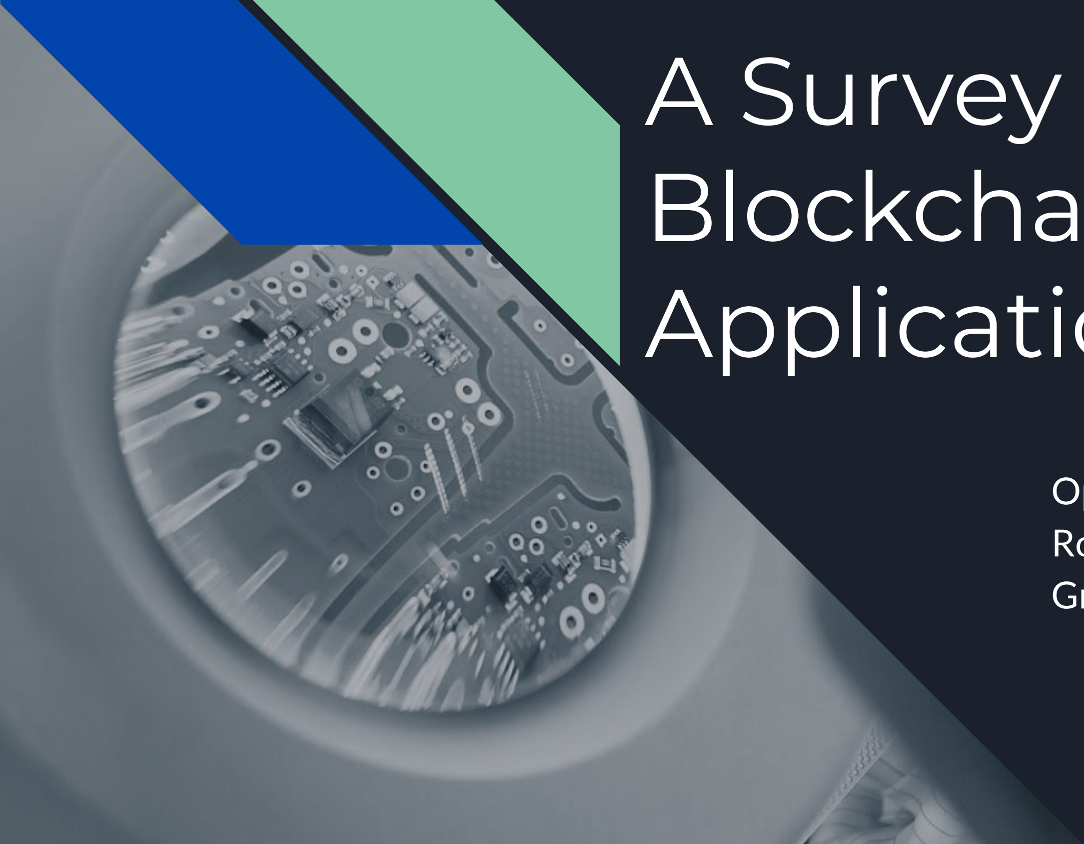
A Survey of Consortium Blockchain and Its Applications



Operating Systems - Class A
Roya Hosseini
Group: OS-striches

Members (OS-striches)

1. Suzal Regmi - Led the literature review on consortium blockchain and identified key OS concepts.

2. Hemang Shimpi - Analyzed layered architecture from the paper; organized insights for slides.

3. Puru Saluja - Researched and explained consensus mechanisms and integrated OS parallels.

4. Rohan Khandelwal - Prepared application use cases, connected class material, polished slides.

Introduction (Context and Background)

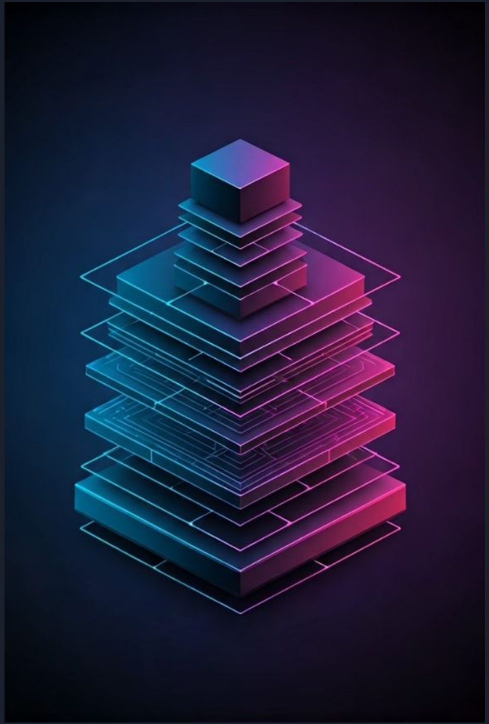
What is Consortium Blockchain?

- A blockchain variant where a set of known, authenticated participants form a semi-decentralized network.
- Unlike public blockchains (e.g., Bitcoin), consortium blockchain participants have permissions, improving trust and control.

Key Idea from the Paper:

- Decompose blockchain into layers: hardware, network, blockchain-core layers (data, consensus, smart contracts), and Layer-II protocols.
- BFT (Byzantine Fault Tolerant) State Machine Replication is central, ensuring that all nodes (like OS processes) converge on a consistent system state.

Introduction (Blockchain as a State Machine)



Blockchain as State Machine Replication (SMR):

- SMR: All nodes replicate the same logical state machine.
- Transactions = state transitions. Once agreed upon, all nodes move from State S0 to S1, etc.
- In OS terms, this resembles synchronized process states where each process (node) maintains an identical “global view,” akin to consistent snapshots of system state.

Introduction (From OS to Blockchain)

Why tie to Operating Systems concepts?

- OS ensures resource abstraction, synchronization, fault tolerance, and concurrency control.
- Consortium blockchain faces similar challenges:
  - Nodes = Processes/Threads
  - Consensus = Advanced Synchronization/Mutual Exclusion (like semaphores but more complex)
  - Resource management (bandwidth, storage) = OS-like allocation to maintain fairness and efficiency.

Motivation and Problem Formulation

Motivation:

- The paper addresses the lack of a comprehensive design overview of consortium blockchains.
- Consortium blockchain is often a better fit for enterprise settings (supply chains, finance) than public blockchains due to improved governance and efficiency.

Problem:

- How to systematically map each architectural layer, identify key challenges (e.g., throughput, scalability, privacy), and highlight solutions across software/hardware boundaries.
- Like OS: need frameworks to handle concurrency, reliability, and resource constraints at scale without trusting all participants.

Method (Layered Architecture Overview)

Layered Approach from the Paper:

- **Hardware Layer:** Physical machines and Trusted Execution Environments (TEEs) for secure code execution.
- **Network Layer:** P2P communication protocols, inspired by OS networking stacks.
- **Layer I (Core):** Data layer (hash chains, Merkle trees), consensus layer (like a global OS scheduler ensuring fairness and correctness), smart contract layer (autonomous “system calls”).
- **Layer II Protocols:** Off-chain scaling solutions, akin to OS caching or virtualization layers, improving scalability and performance without altering the core.

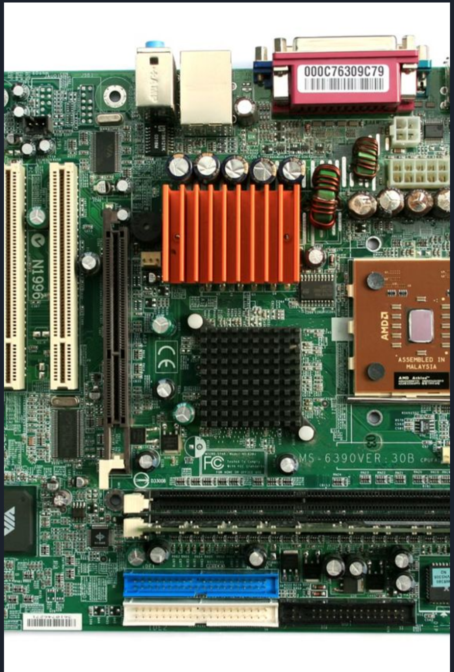
Method (Consensus Mechanisms)

Consensus as the OS “Scheduler”:

- The paper surveys PBFT, Raft, and other BFT mechanisms used in consortium blockchains.
- Like OS schedulers ensuring all processes get CPU time, consensus algorithms ensure all nodes agree on the next block (global state), even with malicious actors.

Uniqueness for Consortium Blockchain:

- Authentication of participants reduces overhead compared to fully public blockchains.
- Deterministic finality, lower latency - much like RTOS (Real-Time OS) scheduling.



Method (Smart Contracts and OS Abstractions)

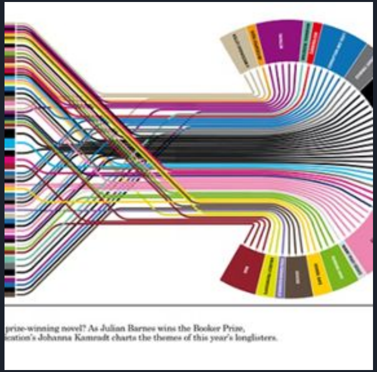
Smart Contracts = OS-level Programs with Guaranteed Execution:

- Pre-agreed logic runs automatically, similar to a protected kernel mode operation ensuring no external tampering.
- Like OS system calls, these contracts interact with shared “state.”
- Enhanced security ensures no single party can alter execution, akin to memory protection in OS.

Results (Performance and Scalability Observations)

Findings from Paper:

- Different consensus models (PBFT vs. Raft) impact throughput and latency.
- Hardware offloads (TEE) and network optimization significantly boost performance.
- Layer-II solutions (channels, side-chains) alleviate on-chain load, similar to OS caching or multi-level scheduling to reduce system bottlenecks.



Unique Insight:

- Tailoring consensus to trust levels is like choosing the right concurrency mechanism in OS: semaphores vs. monitors vs. hardware locks.

Results (Storage and Resource Management)

Data Storage Management:

- Consortium blockchain stores entire transaction history—huge overhead. Paper highlights techniques like Merkle trees and compression.
- Similar to OS file systems optimizing storage: journaling (blockchain immutability), indexing (Merkle proofs), and block size limits (like disk block sizing).

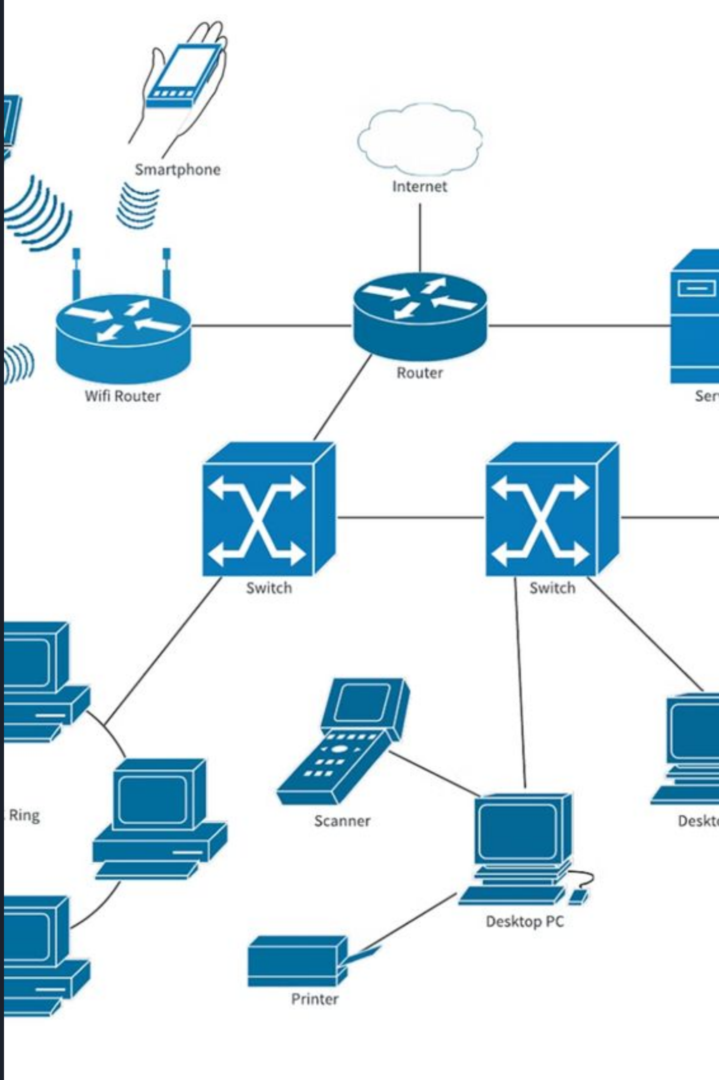
Resource Efficiency:

- Reducing redundant data among nodes can be seen as a distributed OS's challenge of balancing workload and memory usage across machines.

Results (Applications Across Domains)

Applications Mentioned:

- IoT device management: Decentralized trust, akin to OS-level distributed resource tracking.
- Healthcare record sharing: Blockchain acts like a secure OS kernel ensuring integrity and privacy.
- Supply Chain and Agriculture: Trusted traceability, analogous to OS logging and audit trails.
- Smart Grid: Reliable and fault-tolerant power transactions, similar to robust OS handling of critical processes.



References

Primary Paper:

- Chen, X.; He, S.; Sun, L.; Zheng, Y.; Wu, C.Q. “A Survey of Consortium Blockchain and Its Applications.” Cryptography 2024, 8, 12.

Additional References:

- Silberschatz, A.; Galvin, P.; Gagne, G. Operating System Concepts.
- Hyperledger Fabric and Ethereum documentation.
- Relevant OS class syllabus materials.

In conclusion, Consortium blockchain's layered model mirrors OS design (resource abstraction, concurrency control, fault-tolerance). By combining cryptographic integrity with distributed trust, consortium blockchains enable robust applications in multiple domains.

Thank you!