Blockene: A High Throughput Blockchain over Mobile Devices

Sambhav Satija, Apurv Mehra, Sudheesh Singanamalla, Karan Grover
Muthian Sivathanu, Nishanth Chandran, Divya Gupta, Satya Lokam

Microsoft Research India
Trustworthy Transaction Management

• E.g. Financial transactions, property transfers/ownership, etc.

• Traditional model: Use a common, **trusted intermediary**
  • e.g. a Bank (money), Govt. Registry (land ownership), etc.
  • Store current state of the world; **Validate** transactions for integrity

• Sometimes, trusted intermediary is not viable or is cumbersome
  • e.g. Philanthropic donations/NGOs, public spending, intl. bank transfers

• **Blockchains** enable **decentralized** transaction management
  • Multiple parties **store** state; run a consensus protocol for **validation**
  • Robust even if some parties (e.g. < 1/3rd) are malicious
## Blockchain architectures

<table>
<thead>
<tr>
<th>Blockchain Architecture</th>
<th>Applications</th>
<th># Participants</th>
<th>Performance (Trans/sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Public Blockchains (e.g. Bitcoin)</td>
<td>Cryptocurrency</td>
<td>Millions</td>
<td>4 - 20</td>
</tr>
<tr>
<td>Consortium Blockchains (e.g. Hyperledger)</td>
<td>Business processes (e.g. inter-bank payments)</td>
<td>Tens</td>
<td>1000s</td>
</tr>
<tr>
<td>AlgoRand [2017]</td>
<td>Cryptocurrency</td>
<td>Millions</td>
<td>1000s</td>
</tr>
<tr>
<td>Blockene</td>
<td>Citizen-powered audits (e.g. philanthropy)</td>
<td>Millions</td>
<td>~1000</td>
</tr>
</tbody>
</table>
Societal scale applications

- **Philanthropy**
  - Donations to NGOs most common mode: ~USD 10B+/year in India alone
  - **Problem**: Very little transparency/accountability => leakage of funds
  - **Blockchain-based platform for tracking** flow of funds
    - Donor knows exactly what happened to their donation (up to end beneficiaries)

- **Public Spending/e-governance** (e.g. distribution of subsidies)
  - E.g. when intermediaries may be corrupt
Scale & Lightweight Cost: Why important?

• **Scale leads to Security**
  • Central assumption in blockchains: $\frac{2}{3}$rd members are honest
  • Consortium => concentration of power w/ small number of members
    • Can collude/be vulnerable to corruption
  • Shared control of millions of citizens => more collusion-proof

• **Lightweight leads to Scale**
  • Current blockchains need powerful servers: Barrier for adoption
  • For wide altruistic adoption, participation should be almost free
Blockene properties

• **High performance:** 1045 transactions/sec
  • Good enough for real-world applications

• **Large scale:** Millions of participant members

• **Ultra-Lightweight:** Participants only need a smartphone
  • ~60MB/day data usage (**700x cheaper** than alternatives; cellular-data-friendly)
  • MBs of storage (**1000x lower** than alternatives)
  • ~3% battery cost per day (**imperceptible** to user)
Design of Blockene

• A “perfect” democracy

• **Politicians**: “Powerful” & few (100s)
  • Not trusted by citizens
  • Up to 80% politicians can be corrupt
  • *Only execute* citizens’ decisions

• **Citizens**: “Poor” & many (100s of mil)
  • Majority of citizens are honest (> 70%)
  • Collusion is hard because of large numbers
  • *Take all decisions* by majority
    • e.g. whether a transaction is valid,
    • which transactions go into a block, etc.
Division of responsibilities

- **Only trust data signed by “majority” of citizens**

- **Storage** of blockchain
  - Normally, every member stores entire blockchain
  - Blockene: Only politicians store blockchain

- **Communication**
  - Normally, members “gossip” blocks & txns
    - Data intensive; can’t happen w/ mobile phones
  - Blockene: Citizens gossip **through** politicians
    - Citizens read/write to politicians; politicians gossip

- **Compute**
  - Validation / Ordering performed by citizens
Dealing with **corrupt politicians**

- Politicians cannot fabricate messages (need sign by majority citizens)
- **Staleness** Attack
  - Replay old signed messages. E.g. when citizen asks for “latest” block
- **Forget** attack (e.g. drop new block)
- **Fragment world view** attack
- Several new mechanisms
  - Verifiable replicated reads
  - Fan-out writes to "safe sample"
  - Prioritized gossip
- Need only 20% honest politicians
Dealing with “poor” citizens

• Each new block blessed by a random sub-committee of citizens
  • 2000 random citizens out of millions
• Sub-committee is cryptographically chosen
  • Each round/block has a different sub-committee
  • Citizen can prove it’s chosen for a block/round
  • Similar principle to AlgoRand, but more battery-efficient
    • Checking for committee every round kills battery

• Validate w/o storing copy of blockchain
  • Read only state that's needed. Challenging when 80% of "data sources" are malicious
Scalable by design

• Increasing number of citizens by 10x (e.g. 1 million to 10 million)
  • Per-citizen load *reduces* by 10x
  • Per-politician load *nearly constant*: Size of citizen sub-committee is fixed

• Increasing # politicians
  • Per-citizen load *remains constant* ("safe sample" size is fixed)
  • Per-politician load *nearly constant* because of prioritized gossip
Dealing with “greedy” citizens

• What prevents users from spinning up 100s of nodes?
  • Sybil Attacks; Get disproportionate voting power

• Tap into Android trusted hardware (TEE) used for fingerprint auth
  • De-dup Blockene identity w/ public key of TEE (i.e. one identity per phone)

• Can be combined with anonymized real-world identity (e.g. Aadhaar)
  • Blockchain identity tied to a derivative of Aadhaar
Implementation

• Citizen node: Android app
  • Careful threading model to overlap compute + networking

• Politician node: C++ server
  • State-machine based architecture

• ~23k lines of code

• **Evaluated at scale** on Azure across multiple WAN regions
  • Android VM on Azure for citizens
  • Cloud server on Azure for politicians
Evaluation: Transaction throughput

- 200 politician nodes on Azure (8-core, 40 MB/s network b/w)
- 2000 Citizen nodes on Android VMs in Azure (1-core, 1 MB/s b/w)

Spread across 2 Azure regions (WestUS, EastUS)

Committee size independent of # citizens => Performance expected to be same at larger scale

~1045 transactions per sec
Global State Read/Write

Global State stored in Merkle tree at politicians
Citizens need to verify during reads and writes

Naïve: Download challenge-paths for each key (too expensive)
Optimized: Offload compute to politicians; spot-checks with guarantees

<table>
<thead>
<tr>
<th>Config</th>
<th>Upload (MB)</th>
<th>Download (MB)</th>
<th>Compute (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Naïve: GS Read</td>
<td>0</td>
<td>56.16</td>
<td>93.5</td>
</tr>
<tr>
<td>Naïve: GS Update</td>
<td>0</td>
<td>0</td>
<td>93.5</td>
</tr>
<tr>
<td>Optimized: GS Read</td>
<td>0.55</td>
<td>1.6</td>
<td>1.0</td>
</tr>
<tr>
<td>Optimized: GS Update</td>
<td>0.01</td>
<td>3</td>
<td>5.88</td>
</tr>
</tbody>
</table>

242 s/block
12 s/block
Summary

• First known attempt to make blockchains work off mobile phones
  • Resource usage 2 to 3 orders of magnitude lower
• Novel systems design: Citizens + Politicians
• Bridging security vs. performance tradeoff
  • Delegate heavy work to powerful, but untrusted nodes
  • Verify and perform correctness-sensitive work at lightweight nodes
  • Proofs for safety, liveness, and fairness
• Citizens participate in blockchain at negligible cost
• Edge-powered state-management platform