Blockchain deconstructed

Fritz Henglein
University of Copenhagen
henglein@diku.dk

2nd Symposium on Distributed Ledger Technology
Gold Coast, Australia
Related background

- European Blockchain Consortium (ebcc.eu)
- Steering committee chair, Innovation network for Finance IT (CFIR.dk)
- Principal investigator, Functional technology for high-performance architectures (FUTHARK)

Academic background and affiliations

Areas of interest
- Programming language technology
- Theoretical computer science (algorithms, semantics, logic)
- Blockchain technology
- Contract management
- Financial technology
- Enterprise systems
What is a blockchain/DL system?
What is a blockchain/DL system?

Any peer-to-peer decentralized system

• storing a **single, consistent tamper-proof record of events**
• admitting only **appending** of new events
• enforcing a fixed or user-defined protocol (contract) for appending new events
• Guaranteeing **unique ownership** of resources (no double spending)
• with no trusted or privileged parties
• based on **cryptographic** principles
What is a blockchain/DL system?

A computer system characterized by

- organizational and technical **decentralization**;
- **tamper-proof recording** of events and their **evidence**; and
- guaranteed **resource preservation** and **credit limit** enforcement
Organizational and technical decentralization

- **Technical** decentralization: A distributed peer-to-peer system
- **Organizational** decentralization: No single or select group of organizations controls/has privileged rights to system compared to others
- **Governance policy** for regulating membership, functionality, conflict resolution, etc.
  - Group of organizations operating and using system can be open and self-authenticating (nonpermissioned, “distributed ledger technology”) or closed and externally authenticated (permissioned, “blockchain”).
REA accounting model (extended information and contracts)

- **Resource (= asset):** Money, licenses, physical objects (trucks),...
- **Information:** Data, invoices,...
- **Agent:** Person, company, institution, autonomous device,...
- **Contract:** Specification of obligations, permissions and prohibitions
- **Event:**
  - Atomic event:
    - A transfers R to B
    - A transforms R to R’
    - A informs B of I
    - ...
  - Complex event: Set of events that satisfies a given (sub)contract
Tamper-proof recording of events and their evidence

• Event recording: Events are recorded
• Tamper-proof: They cannot subsequently be altered or deleted
• Evidence
  • for atomic events: signature, plus supporting evidence of event having happened (pictures, receipts, DNA samples, GPS data, )
  • for complex events: (mathematical) proof that a set of events is a correct execution of a contract
Digital Twin via physical evidence framework

Physical world

Physical evidence framework

Contract

Alternative state 1

Alternative state 2

History in Blockchain
Guaranteed resource preservation and credit limit enforcement

• **Resource preservation**: Transfers keep the sum of all resources invariant:
  • A transfers 50 ETH to B: The sum of all ETH is the same. Atomically, after event A has 50 ETH less; B has 50 ETH more.
  • We allow negative numbers

• **Credit limit enforcement**: A transfer is only *valid* and *effected* if the credit limits of each agent are respected. For above transfer:
  • If A owns 60 ETH and has credit limit 0: Valid.
  • If A owns 30 ETH and has credit limit 0: Invalid.
  • If A owns 30 ETH and has credit limit 20: Valid.

• No-double-spend guarantee = all agents have credit limit 0.
Use cases for adaptive credit limits

- **Full-reserve monetary system**: One agent (the central bank) has no credit limit (or dynamic credit limit according to some governance regime), all others have credit limit 0.
- **Fractional-reserve monetary system**: A designated set of agents (“banks”) have a dynamic credit limit, all other have credit limit 0.
- **Demand-driven production of physical assets**: A car manufacturer has no credit limit (they produce cars on demand), all others have credit limit 0.
Commutativity theorem, part 1

• **Theorem:** If every agent has an infinite credit limit, then all resource transfers are valid and can be executed in arbitrary order. (Each order results in the same state of ownership.)

• **Corollary:** Contract execution involving $k$ agents requires only consensus by the $k$ agents on which events have happened.

• Note, usually $k=2$. The Internet with TLS (with tamper-proof recording of message sending) is a permissioned blockchain/DL system if there are no credit limits!
Commutative theorem, part 2

• If some agents have finite credit limits, outside validation of their resource transfers is required.
  • Point-to-point communication between the two agents only is *insufficient*.
  • *Some* information about resource transfers must be “leaked” to other nodes for validation.
Canonical distributed ledger architecture (functional view)

- **CLIENT SYSTEM:** SINGLE PARTY INFORMATION
  - subscribe
  - private messages
  - post
  - query

- **SHARED SYSTEM:** RESOURCE TRANSFER VALIDATION ONLY

- **CLIENT SYSTEM:** SINGLE PARTY INFORMATION
  - subscribe
  - private messages
  - post
  - query
Canonical distributed ledger architecture
(distributed systems view)
Conclusions and open problems

• No consensus on globally total order of events is necessary
  • Current blockchain/DL systems solve an unnecessarily hard problem (transaction order consensus)

• Not even consensus on partial order of events is strictly necessary.

• Consensus on resource transfers needs to ensure that the set of all eventually validated resource transfers respects all credit limits.

• Specialized consensus protocols for resource transfers are conceivable and needed for scalability:
  • Hierarchical clearing and settlement (hierarchical “sharding” by partitioning of agents)
  • Time- and resource-sensitive validation (bigger transfers require more time)
  • Insurance (applying transaction fees to covering losses due to overdrafts detected too late)
  • ...
Thank you!