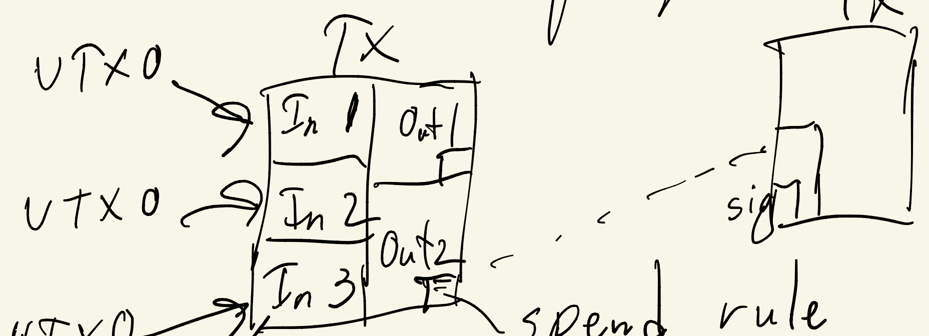
CS-438
Decentralized Systems Engineering

Week 12

Smart contracts

- Bitcoin: "pay to Script"



authorization/ proof (simple: sig)

⇓

(gen: arbitrary binary blob input to script)

spend rule (simple: pub key)

⇓

(gen: script) program to check validity of a spend req → Y/N

Who runs the code?

- user who wants to spend
- all miners run when validating TX for block
  - first to create block
  - then to valide block

Determinism!

Miners need to achieve consensus on block validity - must not disagree

Termination: bound miner effort

- Bitcoin: no (backward) branches / loops
- Ethereum: explicit "gas" bound
  - if gas runs out?
  - reverts the state to maintain consistency
  - miner keeps the gas payment TX still included in ledger

Applications

- Trustless insurance (AXA "Fizzy" - flight delay ins.)
- New payment/finance methods, settlements, ...
- Decentralized naming (DNS, NameCoin)
- Tokenization (ICO - initial coin offerings)
- Storage - on-chain (expensive) or to manage off-chain storage
- Programmable markets: auctions, prediction markets, quadratic voting
- Games (gambling, ...) Crypto-Kitties
- Decentralized online governance/autonomous organizations (DAOs)

Issues & limitations

- Inefficiency of (deterministic) VM - example (partial) solution: eWASM
- Input problem (Oracle problem) - trusted authority
    - emerging: decentralized oracles (voting)
- Smart contract bugs ("The DAO")
    - recourse / recovery mechanisms?
- Front-running attacks ("Dark Forest")
- Can't keep secrets
    - keep secrets off-chain, zk-proofs
    - on-chain secrets (Calypso)
- Improvements / evolution difficult
    - permissionless innovation?