Ethereum

Bitcoin

- We look at the Bitcoin idea as a state transition system
  - There's a state consisting of the ownership status of all existing Bitcoins
  - Each new block is a set of transactions that transforms the state and moves around Bitcoin
- The state is a collection of all coins (unspent transaction outputs or UTCO) that have been minted and not spent
  - A transaction contains 1+ inputs that corresponds to an existing UTXO and an output with all new UTXO
  - In Bitcoin, each UTXO has a corresponding owner (its public key)
  - When doing a transaction, we make sure that all UTXO have not been sent yet
- The main point of Bitcoin is to make sure that all of these transactions appear in the same order
  - The only way to attack the Bitcoin network would be to try to try to void sending money to people by trying to get something that invalidates that sending later by putting an out of order entry in
- We could try implementing scripting straight into Bitcoin with no modifications:
  - UTXO in Bitcoin is owned by a script that takes in data and is verified iff the script is valid
  - Limitations:
    - Not Turing-complete: lack of loops
      - This is because we could then have infinite loops during transaction verification
    - Value-blindness
      - No fine-grained control over amount that can be withdrawn because UTXO are all or nothing
    - Lack of state
      - UTXO can either be spent or unspent so there is no way to store state in contracts
    - Block-chain blindness
      - UTXO are blind to data such as the nonce / timestamp / previous blocks so there is no way to use that to generate RNG

Ethereum

- The state of Ethereum is made up of accounts, each of which contains:
  - The nonce, a counter used to make sure each transaction can only be processed once
  - The account's current ether balance
  - The account's contract code (if it exists)
  - The account's storage (empty by default, contains kv pairs)
- Two types of accounts:
  - Externally owned account: controlled by private keys and have no code
  - Contract accounts: controlled by their contract code
- One sends messages from an externally owned account by creating and signing a transaction
- In a contract account, every time it receives a message, its code activates
  - This allows it to write to internal storage and send other messages / create contracts
- Contracts are not something that fulfilled or complied with but instead are autonomous agents
  - They always execute a specific piece of code when "poked" by a message
- Transactions are sent from an externally owned account and contain:
  - The recipient of the message
  - A signature identifying the sender
  - The amount of ether to transfer
  - An optional data field
  - A STARTGAS field, representing the max number of computational steps the transaction execution is allowed to take
  - A GASPRICE field, representing the fee the sender pays per computational step

    - The latter two are used to prevent denial of service due to infinite loops
- Messages are sent from contracts to other contracts and contain:
    - The sender of the message
    - The recipient of the messages
    - The amount of ether to transfer
    - An optional data field
    - A STARTGAS value
- The gas allowance by a transaction / contract applies to total gas consumed by that transaction and all sub-executions

- Ethereum blocks are mined much faster at 12 seconds per block because of the GHOST protocol

Lecture

- Many applications are centralized where we have to trust the developers
    - For instance, there could be a casino that just lies about the results
    - Open source doesn't really help because there's no way to tell if the given source code is actually being run
- We want to have some distributed, trustless authority that anyone can look inside of to see what's happening
    - To make the economics of this work out, it has to have a built-in currency system
- Smart contracts are like classes in that you can instantiate them and then have other people interact with them
    - All operations that perform work cost gas to fuel the compute power used to run this operation on Ethereum
    - Get operations don't cost gas since they don't need to actually go onto the blockchain

---

- Ethereum state is a mapping from addresses to either:
    - Externally owned accounts with a balnce
    - Contract account with a balance, code, and state
- Actions on the state machine either:
    - Deploy contracts with code and initial value
    - Message with address, method to invoke, data, and ethereum value