Ethereum training course

Note: This document is scoped to general development of smart contracts on the Ethereum platform. It does not cover the installation and configuration of software in a rapidly-changing ecosystem. The target audience is for developers who are interested in building on a blockchain platform.

What is a blockchain?

Components of a distributed immutable ledger

- Asymmetric encryption (https://en.wikipedia.org/wiki/Public-key_cryptography)
    - Signing & verifying
    - Encryption & decryption
- Merkle data tries (https://en.wikipedia.org/wiki/Merkle_tree)
    - Child node points to parent node
    - Allows for efficient verification of large data structures
- Peer-to-peer network (https://en.wikipedia.org/wiki/Peer-to-peer)
    - Networking component
    - Helps ensure resiliency
    - DHT (https://en.wikipedia.org/wiki/Distributed_hash_table)
- Consensus Algorithm (https://en.wikipedia.org/wiki/Consensus_(computer_science))
    - EthHash Proof-of-Work Mining (http://ethdocs.org/en/latest/mining.html)
    - Set chronological ordering of transactions
    - Settlement of information providing gradual finality

What is Ethereum?

Blockchain with smart contract

- History of development (https://blog.ethereum.org/2016/02/09/cut-and-try-building-a-dream/)
    - White paper released by Vitalik Buterin in December 2013
    - Announced in Miami, FL, USA in January 2014
    - Crowdsale June August 2014
    - Public chain released June 2015
- How does it differ from Bitcoin?
    - Bitcoin is a simple ledger for tracking account balances
    - Ethereum is a generalized state machine similar to a computer
    - Anyone can publish new software to the Ethereum blockchain
    - Pros and cons of Ethereum balances vs. UTXOs (https://ethereum.stackexchange.com/questions/326/what-are-the-pros-and-

cons-of-ethereum-balances-vs-utxos/386#386)

- How it's the same
    - Satoshi blockchain with economic incentivization
    - Proof-of-work consensus algorithm

Ethereum Virtual Machine (EVM)

- custom stack-based virtual machine (https://ethereum.stackexchange.com/questions/119/what-opcodes-are-available-for-the-ethereum-evm)
- Sandboxed execution environment
- Deterministic outcome
- Stores smart contract code and data
- Prosesses the interaction of an incomming transaction to a smart contract

Building Ethereum Decentralized Applications (dApps)

Writing Smart Contracts

- High level language Solidity (https://solidity.readthedocs.org/en/latest/) compiles to EVM code
- Write synchronous code that runs above the consensus mechanism
- Only worry about handling transactions, not forks or reverts

Events for notification

- Record meaningful events (e.g. new registration)
- Search for past events or listen in real time
- Helpful for building live-updating UIs (https://github.com/ethereum/wiki/wiki/JavaScript-API#web3ethfilter)

Writing dApp Frontends

- Similar to normal web app
- Official Javascript library web3.js (https://github.com/ethereum/wiki/wiki/JavaScript-API)

ABI for external interaction

- "Abstract Binary Interface" or "Application Binary Interface"
- ABI (https://ethereum.stackexchange.com/questions/234/what-is-an-abi-and-why-is-it-needed-to-interact-with-contracts) explains how to interact with Smart Contract binary code
- Contains a list of the contract's methods and properties
- Allows for encoding/decoding of data in other languages, such as JavaScript

Transaction validation

Economic guarantees

- Validators are assumed to act greedily

- Incentivised to process transactions with fees and protocol subsidy ("block reward")

Proof of Work

- Longest versus heaviest/GHOST
- Backed by value of capital investment & electricity expenditure

Proof of Stake

- Consensus by bet
- Dynamic finality ("wait as long as you want")
- Backed by value of all bets

Ether native token

- What is Ether? (http://ethdocs.org/en/latest/ether.html)
- Why is a native token neccessary?
    - Incentivize transaction validation
- What are the subunits of ether?
    - Wei is base unit

Gas

What is Gas? (http://ethdocs.org/en/latest/contracts-and-transactions/account-types-gas-and-transactions.html#what-is-gas)

- A resource cost of the EVM
- To prevent spam & tragedy of the commons (https://en.wikipedia.org/wiki/Tragedy_of_the_commons)

Why is gas required?

- Different operations require differing amount of resources
- Car analogy
    - More fuel consumed when going uphill versus coasting downhill

How is gas paid for and consumed?

- Senders must pay for all gas consumed
- Price per gas is expressed in native token, Ether (i.e. 0.00000002 ether/gas)
- Miners/validators can include/exclude transactions at their preference

Scalability

How to Increase transactions/sec

- Shards to allow for parallel processing (not all nodes need to validate each transaction)

How to decrease Blockchain size

- More complex VM requires more storage space (https://ethereum.stackexchange.com/questions/521/what-does-it-mean-to-run-code-on-the-blockchain-wouldnt-blockchain-become-hu)

Light clients

- Different from archive/full nodes
- Only store portions of chain relevant to them
- Less validation, but reliance on multiple peers reduces malfesense

Permissioned chains

Usage

- First blockchain is public goods network
- As coopreration mechanism, this is where it excels most

Private is an ambiguous term

- "Private" chains are probably internal/consortium/federated
- Limited validator set
- Limited data readability?

Data privacy

Hiding data on public ledgers

- No intrinsic solutions (http://ethdocs.org/en/latest/frequently-asked-questions/frequently-asked-questions.html#can-i-store-secrets-or-passwords-on-the-ethereum-network)
- External encryption
    - Use PGP or similar to encrypt/decrypt external to the blockchain
- Zero knowledge proofs? (https://en.wikipedia.org/wiki/Zero-knowledge_proof)

Storing large files

- Blockchain expensive & inefficient for storing BLOBs
    - BitTorrent/Kademlia more suited to non-consensus information
- IPFS (https://ipfs.io/)/Swarm (http://ethdocs.org/en/latest/contracts-and-transactions/developer-tools.html#swarm) natural evolution for blockchain world
    - Storing content hashes in the blockchain is ideal hybrid approach

Epherimal communication

Whisper

- What is Whisper (http://ethdocs.org/en/latest/contracts-and-transactions/developer-tools.html#whisper)
- Development & delivery (https://ethereum.stackexchange.com/a/388/43)

Designing DApps from scratch

- Separate logic from presentation
- Codify business logic as smart contracts
- Apply front-end UI/UX through static HTML/JS/CSS
- Deploy

Ethereum tools

Development frameworks

- Truffle (https://truffle.readthedocs.org/en/latest/)
- Embark (https://iurimatias.github.io/embark-framework/)
- Using Meteor (https://github.com/ethereum/wiki/wiki/Dapp-using-Meteor)

IDEs

- Visual Studio via plugin (https://visualstudiogallery.msdn.microsoft.com/96221853-33c4-4531-bdd5-d2ea5acc4799)
- Mix (https://github.com/ethereum/webthree-umbrella/releases/latest)
- Browser-Solidity (https://chriseth.github.io/browser-solidity/)

Chain explorers

- etherscan.io (https://etherscan.io)
- etherchain.org (https://etherchain.org)
- ether.camp (https://live.ether.camp/)

Appendix

APIs and reference docs

- JSON-RPC (https://github.com/ethereum/wiki/wiki/JSON-RPC)
- Web3 JavaScript API (https://github.com/ethereum/wiki/wiki/JavaScript-API)
- Solidity documentation (https://solidity.readthedocs.org/)
- EVM Yellow Paper (http://gavwood.com/Paper.pdf)

More resources

- Ethereum Whitepaper (https://github.com/ethereum/wiki/wiki/White-Paper)
- Ethereum Project website (https://ethereum.org)
- Ethereum Blog (https://blog.ethereum.org)
- Ethereum GitHub (https://github.com/ethereum)
- Ethereum Meetups (http://ethereum.meetup.com/)
- Ethereum Stack Exchange (http://ethereum.stackexchange.com/)
- Ethereum Reddit (http://www.reddit.com/r/ethereum)
- Ethereum Network Status (https://ethstats.net/)

- Ethereum community documentation (http://ethdocs.org)

Acknowledgements

Prepared by Aaron Davis & Taylor Gerring on behalf of Wanxiang Blockchain Labs