Quorum:

Ethereum for enterprise applications

jpmorgan.com/quorum

ELI5: Quorum

Project Goals

Create a **permissioned** version of Ethereum that supports:

✓ **Governance** - nodes & activity are tied to real world identities

✓ **Confidentiality** - details of transactions are private

✓ **Security** - no trust is assumed between participants / nodes

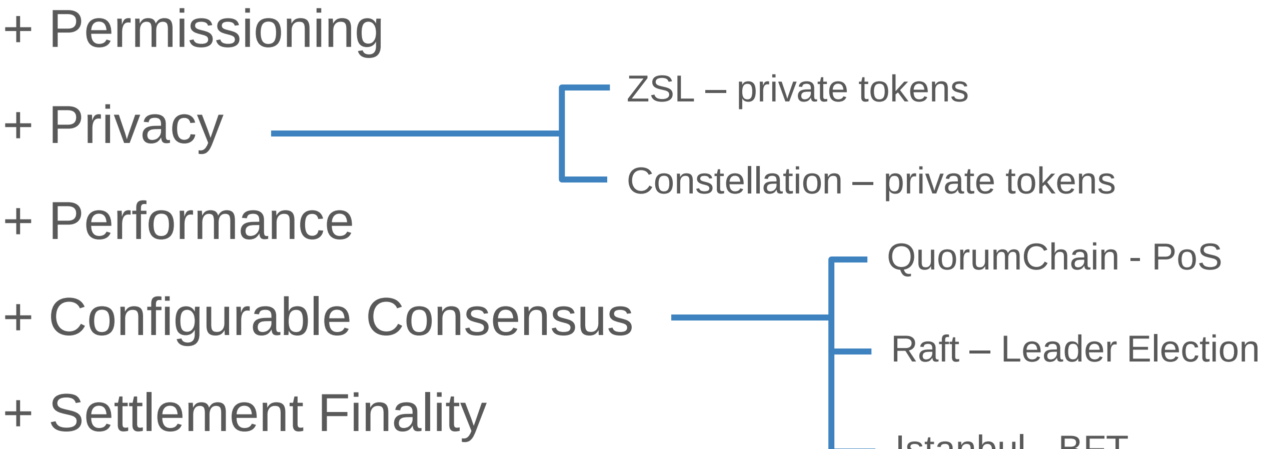
Stay as close as possible to the public Ethereum codebase

Work with the open source community and help define standards

Build a platform that can run in an enterprise production environment

ELI5: Quorum

ethereum



+ Permissioning

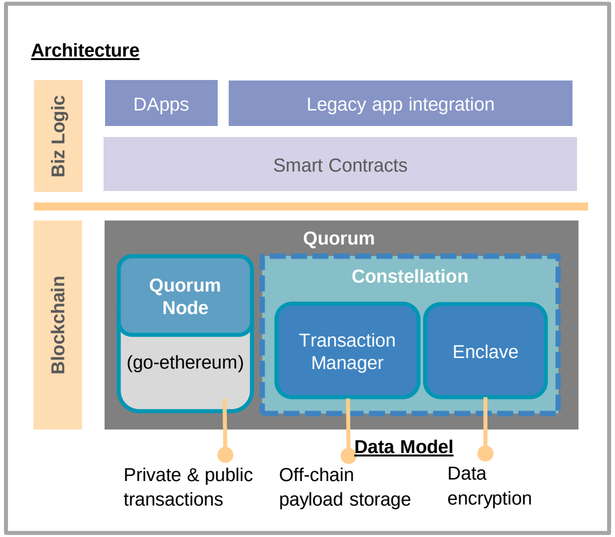
+ Privacy
  - ZSL – private tokens
  - Constellation – private tokens

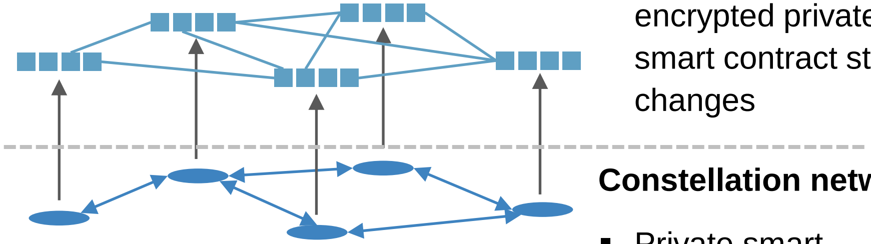
+ Performance

+ Configurable Consensus
  - QuorumChain - PoS
  - Raft – Leader Election
  - Istanbul - BFT

+ Settlement Finality

ELI5: Quorum





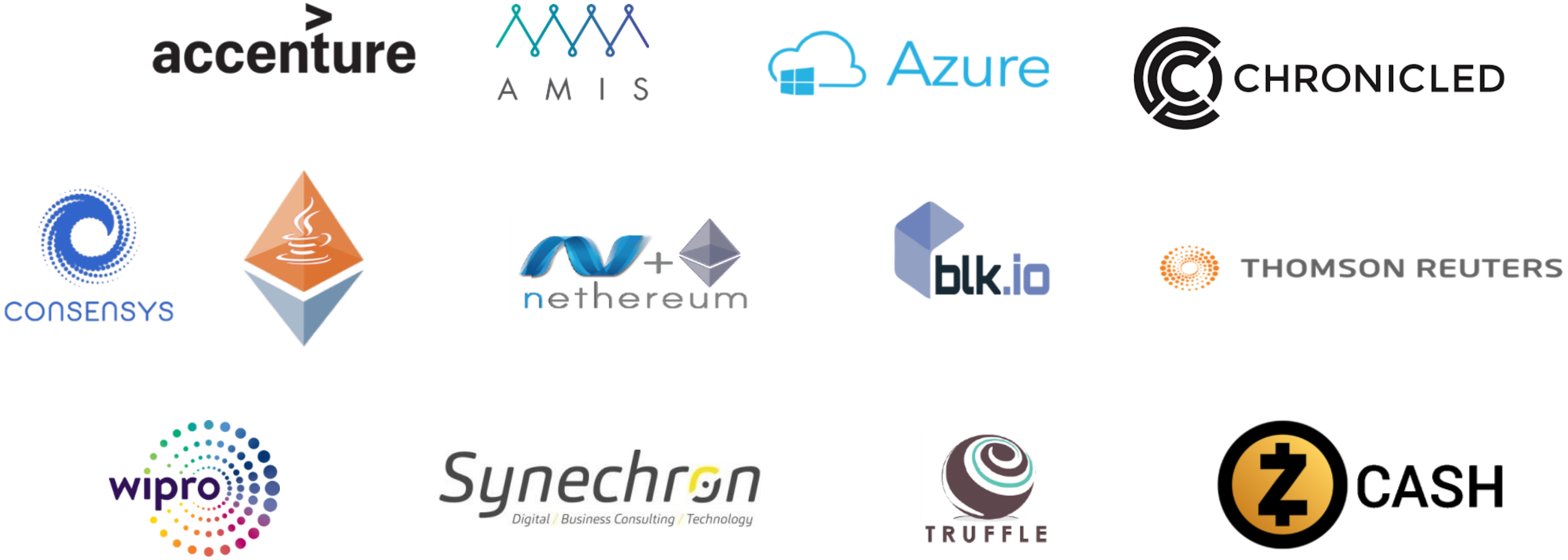
Shared blockchain

- Stores vanilla Ethereum transactions as well as hashes of encrypted private smart contract state changes

Constellation network

- Private smart contracts transmitted point-to-point so that only relevant parties receive them
- Keys communicated peer-to-peer

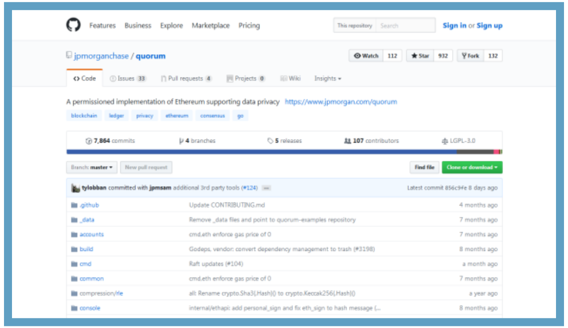
An ecosystem is emerging



Learn more, get involved!

What is Quorum?

Enterprise-ready distributed ledger and smart contract platform



jpmorganchase / quorum

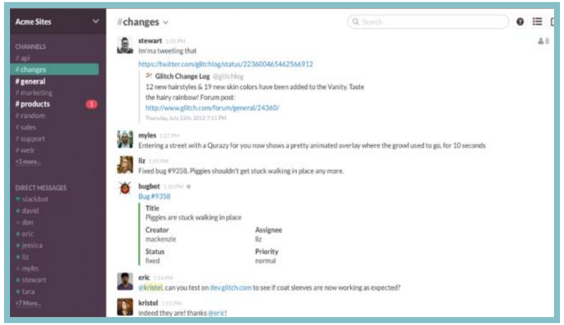
A permissioned implementation of Ethereum supporting data privacy https://www.jpmorgan.com/quorum

blockchain ledger privacy ethereum consensus go

7,864 commits · 4 branches · 5 releases · 107 contributors · LGPL-3.0

tylobban committed with jpmsam additional 3rd party tools (#124) … Latest commit 856c9fe 8 days ago

.github	Update CONTRIBUTING.md	4 months ago
_data	Remove _data files and point to quorum-examples repository	7 months ago
accounts	cmd.eth enforce gas price of 0	7 months ago
build	Godeps, vendor: convert dependency management to trash (#3198)	8 months ago
cmd	Raft updates (#104)	a month ago
common	cmd.eth enforce gas price of 0	7 months ago
compression/rle	all: Rename crypto.Sha3{,Hash}() to crypto.Keccak256{,Hash}()	a year ago
console	internal/ethapi: add personal_sign and fix eth_sign to hash message (…	8 months ago



stewart 1:01 PM
Im'ma tweeting that
https://twitter.com/glitchlog/status/223600465462566912
Glitch Change Log @glitchlog
12 new hairstyles & 19 new skin colors have been added to the Vanity. Taste the hairy rainbow! Forum post: http://www.glitch.com/forum/general/24360/
Thursday, July 12th, 2012 7:11 PM

myles 1:07 PM
Entering a street with a Qurazy for you now shows a pretty animated overlay where the growl used to go, for 10 seconds

liz 1:09 PM
Fixed bug #9358. Piggies shouldn't get stuck walking in place any more.

bugbot 1:10 PM
Bug #9358
Title
Piggies are stuck walking in place
Creator
mackenzie
Assignee
liz
Status
fixed
Priority
normal

eric 1:14 PM
@kristel, can you test on dev.glitch.com to see if coat sleeves are now working as expected?

kristel 1:15 PM
indeed they are! thanks @eric!

Splash page jpmorgan.com/quorum

Code github.com/jpmorganchase/quorum

Slack quorumslack.azurewebsites.net/

Quorum

Announcing:

Quorum + ZSL

jpmorgan.com/quorum

zk-SNARKs: the cutting edge of cryptographic privacy

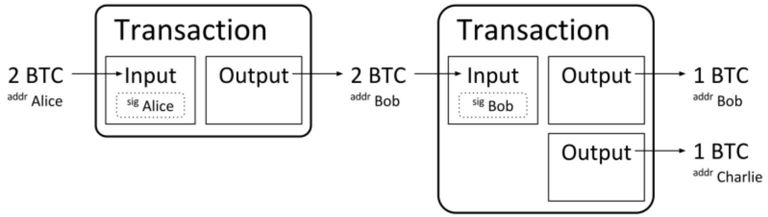
zk-SNARKs allow verification of the correctness of computations without having to execute them, without even learning what was executed (just that it was done correctly)

- Helps address issues with transparent token fungiblity (e.g. Bitcoin)
- May eventually allow private smart contracts to run on the public blockchain

Ethereum integration already underway

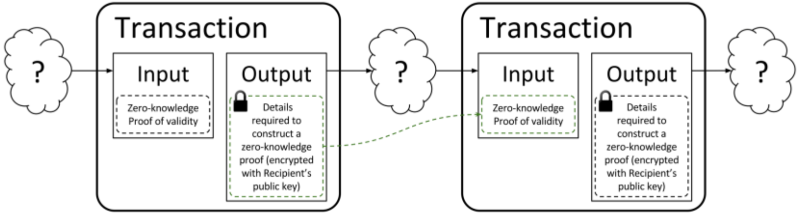
- “Baby ZoE” - anonymous sending of Ether tokens (written for Parity)
- Project Alchemy – decentralized exchange between Ethereum & Zcash

Transaction verification in Bitcoin



2 BTC
addr Alice
Transaction
Input
sig Alice
Output
2 BTC
addr Bob
Transaction
Input
sig Bob
Output
1 BTC
addr Bob
Output
1 BTC
addr Charlie

Transaction verification in Zcash



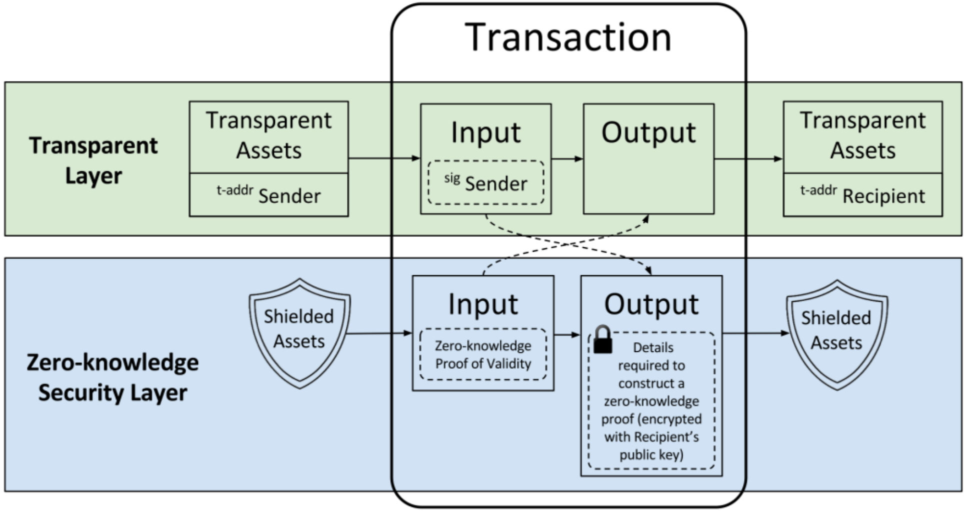
?
Transaction
Input
Zero-knowledge Proof of validity
Output
Details required to construct a zero-knowledge proof (encrypted with Recipient’s public key)
?
Transaction
Input
Zero-knowledge Proof of validity
Output
Details required to construct a zero-knowledge proof (encrypted with Recipient’s public key)
?

ZSL: zk-SNARKs for the enterprise

ZSL enables transfer of digital assets on a distributed ledger **without revealing any information about the sender, recipient, or quantity of assets**, while ensuring that:

- Sender is authorized to transfer ownership of the assets in question
- Assets have not been spent previously (i.e. prevention of double spend)
- Transactions inputs equal its outputs (mass conservation)

Protip: [ZSL is to Zcash] as [Blockchain is to Bitcoin]



Quorum R&D: Hybrid privacy design

Constellation:

- Quorum’s “privacy engine”
- Allows private smart contract execution
- No encrypted data is stored on the shared blockchain
- Private contracts are stored locally
- Hashes of encrypted private contracts are stored on the shared blockchain, ensuring system-wide integrity
- Uses public/private key encryption
- Prevention of double spend requires specific application architectures

Zero Knowledge Security Layer:

- A Zero-Knowledge Security Layer:
  - Can be layered on top of any distributed ledger solution
  - Can be integrated with any consensus mechanism
- Makes mass conservation and prevention of double spend possible for shielded tokens without compromising the decentralized nature of the ledger
- Uses zk-SNARKs (zero-knowledge cryptographic proofs)
- For ZSL on Ethereum, smart contracts must still be executed in the clear