QUORUM

A permissioned implementation of Ethereum supporting data privacy

September 2016

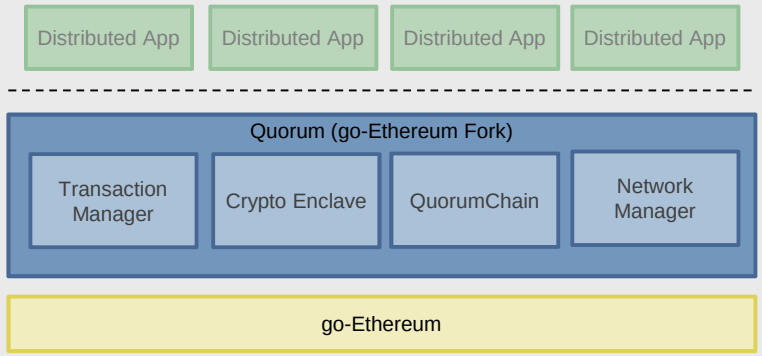
J.P.Morgan

Quorum: A permissioned implementation of Ethereum supporting data privacy

Highlights

- **Built on Ethereum**
  - First mover advantage. In production since July, 2015.
  - 50,000+ unit tests, Security Audits, Bounty Program
  - Largest Ecosystem of Developers, Tools, DApp's
  - Public Ethereum blockchain protects over $1B+ Ether[1]
- **Simple Privacy Design**
  - Supports both private and public transactions and smart contracts
- **Single Blockchain Architecture**
  - All public and private smart contracts and state derived from a single, common, complete blockchain of transactions validated by every node in the network
  - Private smart contract state validated by parties to contract only
  - Best of both worlds… every node validating the list of transactions while only exposing details of private transactions and contracts to relevant parties
- **High Performance**
  - Able to process **dozens to hundreds of transactions per second**, depending on system configuration; enough to support institutional volumes

Architecture



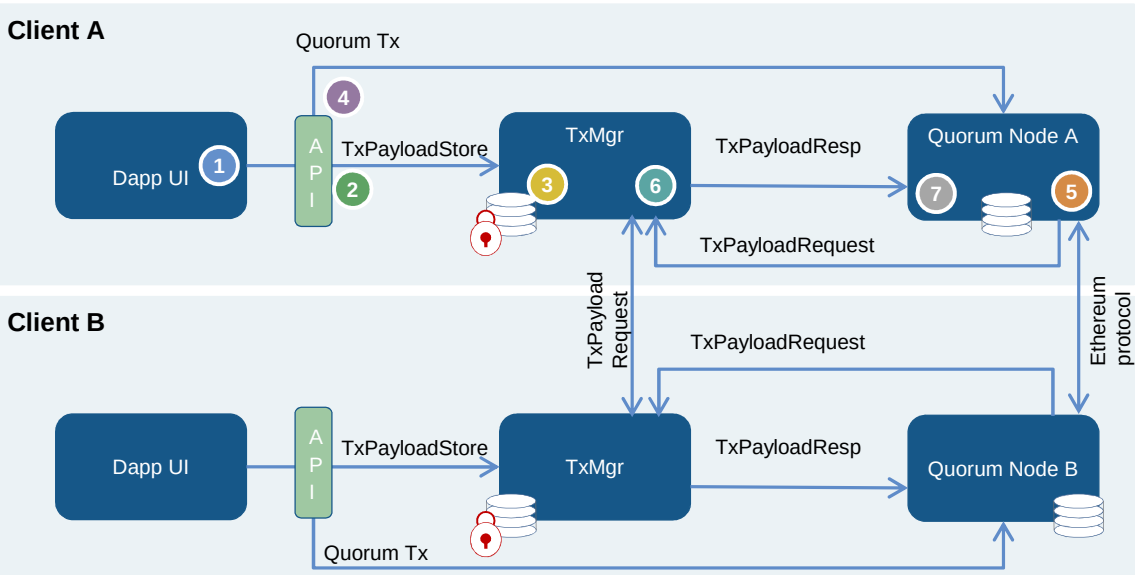
Components

- **Transaction Manager** – allows access to encrypted transaction data for private transactions, manages local data store and communication with other Transaction Managers
- **Crypto Enclave** – responsible for private key management and encryption and decryption of private transaction data
- **QuorumChain** – voting-based, BFT-hardened consensus mechanism that utilises core Ethereum features to verify and propagate votes through the network
- **Network Manager** – controls access to the network, enabling a permissioned network to be created

[1]As of 22-Sep-2016

A pragmatic approach to privacy

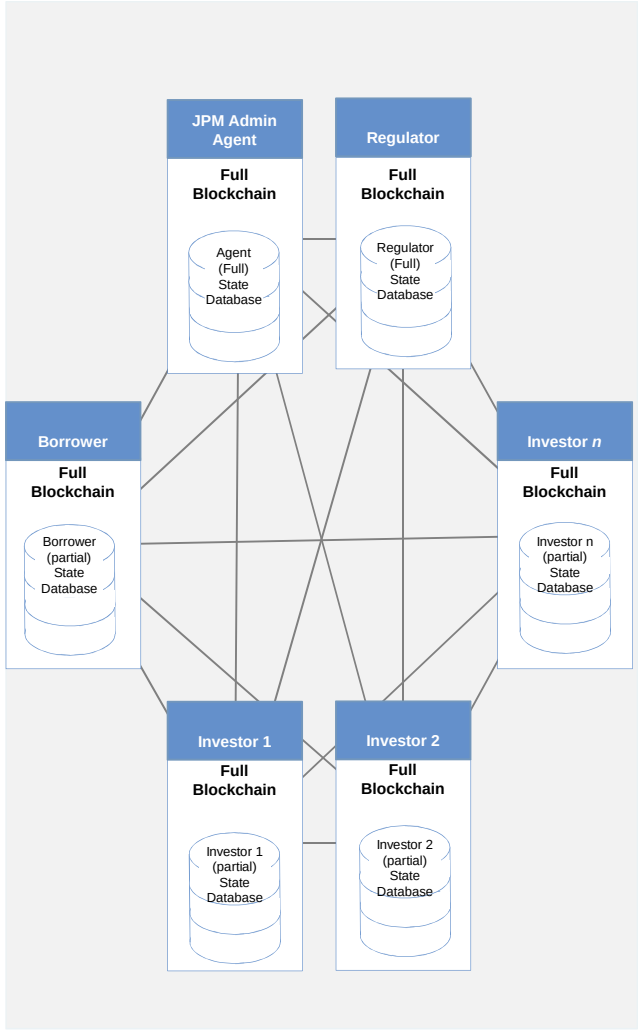
Simple Privacy Design



1. Dapp sends transaction to Quorum Node, specifying recipient and transaction payload
2. **Prepare Tx Payload Record** by generating a symmetric key, encrypt the payload with symmetric key, hash the encrypted payload, encrypt the symmetric key with the public keys of the parties to the Tx, then send to the TxMgr for storage.
3. **TxMgr** validates the sending signature and stores the TxPayload message
4. **Tx sent** to the Quorum node containing only the hash of the encrypted payload generated in step 2.
5. **Quorum Node** receives a new block for validation containing the private Tx. It requests the payload data from the TxMgr (passing its Pubkey, TxHash, Sig).
6. **TxMgr** validates the signature, looks up the TxHash and if the requester is party to the Tx, return the encrypted payload and encrypted Symmetric key.
7. **Quorum Node** decrypts the symmetric key, decrypts the Tx Payload and sends to the EVM for contract code execution.

TxPayload includes:
- Encrypted Tx payload
- Hash of encrypted Tx payload (TxHash)
- Party 1 Public Key encrypted Symmetric Key
- Party 2 Public Key encrypted Symmetric Key
- Party *n* Public Key encrypted Symmetric Key

Full Blockchain, Partial State dB



The future of Quorum

Details….

Forward compatibility

- Quorum is built in partnership with **EthLab (Jeff Wilcke co-founder)**
- Quorum is a minimalistic fork of the Go Ethereum client and will be updated in line with future Ethereum releases
- As Quorum is a derivative of Go Ethereum, it is licensed under GPL/LGPL. Alternative implementations could be licensed differently

Next Steps:

- Open Source codebase & toolkits
  - Quorum platform
  - SDK & Reference Application
- Obtain feedback, iterate, collaborate
- Continue build out of product:
  - Pluggable consensus
  - Further performance optimizations

Contacts

Brian Marchiony
Head of CIB Marketing & Communications
brian.j.marchiony@jpmorgan.com

Amber Baldet
Program Lead, Blockchain Center of Excellence
amber.baldet@jpmorgan.com

David Voell
Engineering Lead, CIB Emerging Technologies
david.l.voell@jpmorgan.com