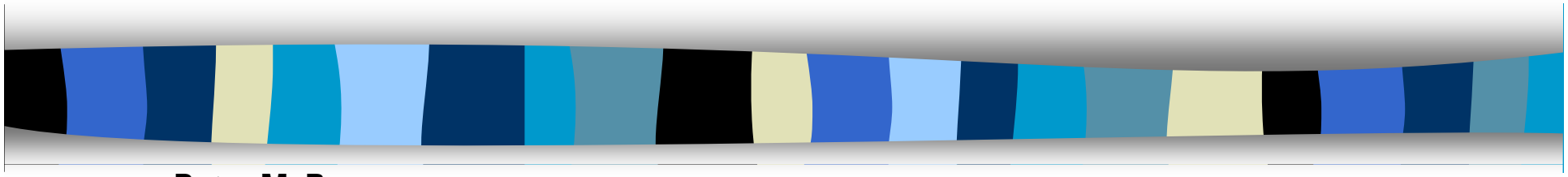
7CCSMDLC: Distributed Ledgers & Cryptocurrencies

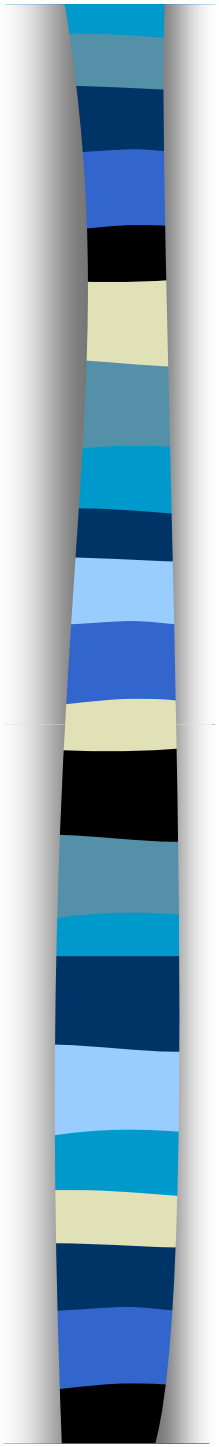
Lecture 6: DLT Infrastructure & Platforms



Peter McBurney
Professor of Computer Science
Department of Informatics
King's College London

Email: peter.mcburney@kcl.ac.uk
Bush House Central Block North – Office 7.15

2020



Outline

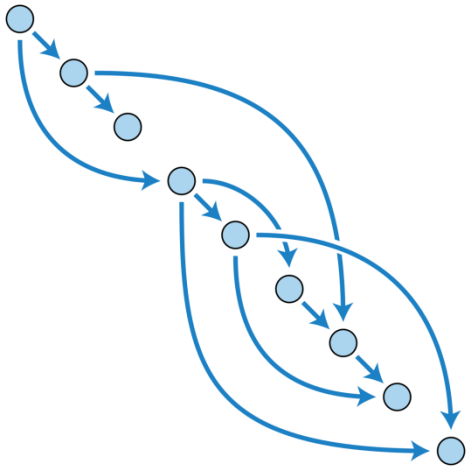
- Distributed Ledgers
- Ethereum
- Corda
- Hyperledger
- Others
- Case Study

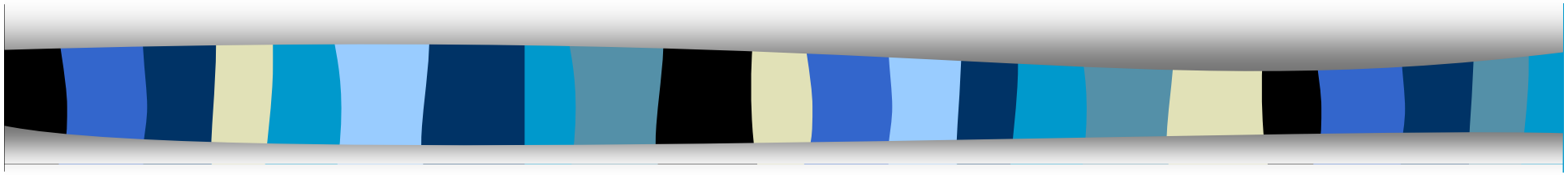
Distributed Ledgers

- Bitcoin Blockchain was designed for a specific use-case
  - Decentralized electronic currency
  - Key issue: preventing double-spending when no one is in control
- However, it opened our eyes to a whole new class of technology
  - Technologies where transactions are witnessed & validated
  - Participants share state of certain variables
  - Transactions are linked together to strengthen resistance to attack & fraudulent revision
- Key elements:
  - P2P (decentralized) data sharing
  - Rule-governed processes to enable interactions between entities lacking mutual trust.

Which features are essential?

- Open-ness?
  - Permissioned ledgers
- Having an electronic currency?
- Having particular protocols or consensus mechanisms?
- Having blocks or chains?
  - Can have a hash-graph (entries linked to past entries by hashing)
  - DAG - Directed Acyclic Graph (see image)
    - Finite directed graph with no cycles
- A world of new structures still being explored.



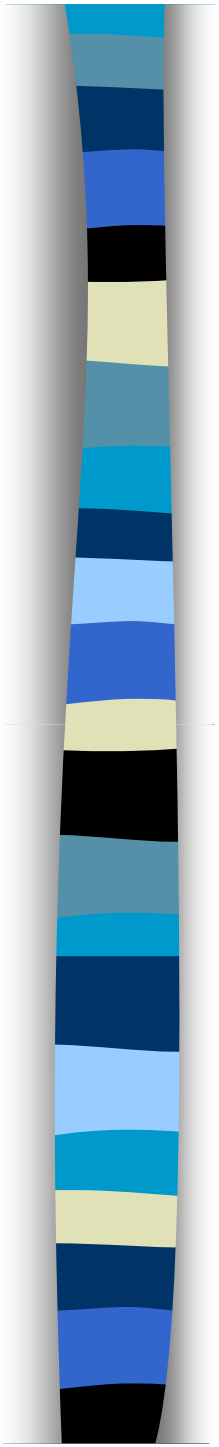


Ethereum

Ethereum

- Language for Bitcoin Blockchain (Script) is limited
  - Desired a blockchain with a full programming language
- Ethereum
  - Blockchain platform with a full language Solidity
  - Turing-complete language (so able to do loops)
- 2013: Proposed by Vitalik Buterin
- July-August 2014: ICO
  - USD$18.4 million
- 30 July 2015: System live

- Ethereum Foundation
  - HQ in Zug, Switzerland
- Currently: About 8K nodes.



Ethereum Features

- Ethereum is a public (open) distributed ledger with an external cryptocurrency Ether
  - ETH or Ξ (Xi)
  - Code now on Muir Glacier (released 2020-01-01, at block 9,200,000)
  - Called Ethereum MainNet
- Uses proof-of-work
  - But plans to switch to Proof-of-Stake
  - Still imminent (delayed several times)
  - Both protocols will operate for a transition period
- Main intended application: programmable DL (smart contracts)
- Imagines a state-transition machine
  - The machine exists on all the full nodes of the network
  - Where programs (smart contracts) act to change the state of the machine.

Ethereum vs Bitcoin

- Blocktime
  - Bitcoin: 10 minutes
  - Ethereum: 15 seconds
- Mining of new coins:
  - BTC halves every 4 years
  - Ether generates new coins at a constant rate
- Total number of currency:
  - BTC: 21 million hard cap
  - ETH: No hard cap (although this may change in the future)
    - Currently: Ξ 110 million (February 2020)
    - Risk of inflation
- Size of network
  - Bitcoin (started 2009): 10,543 nodes
  - Ethereum (stared 2015): 8,245 nodes.

Ethereum Gas

- Ethereum separates cryptocurrency from measurement of work done

- Gas cost – unit of measurement for transactions
  - eg, 6 gas for each 256-bit hash
  - Like KiloWatts (units of measurement of electricity)
  - Based on complexity of processing, bandwidth needed, memory usage
  - The more complex the commands, the more gas you need to offer

- Gas price (measured in ETH)
  - How much initiator is willing to pay for the transaction to be processed

- Initiator of transaction specifies in ETH the price he/she is willing to pay per gas and the total number of gas permitted to be used

- Total processing fee in ETH =

  Gas limit (total # of gas to be used) X gas price in ETH.

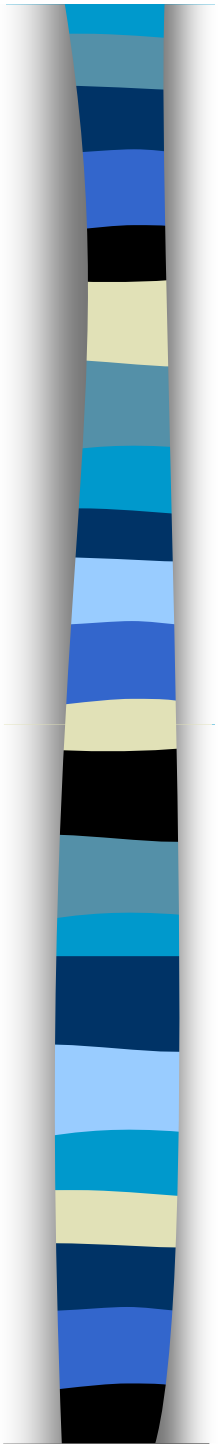
Examples of gas costs

step	1	Default amount of gas to pay for an execution cycle
stop	0	Nothing paid for the suicide operation
sha3	20	SHA3 Hash
memory	1	For each additional word when expending memory
tx	500	Paid for every transaction

- Miners can accept proposed gas price or not
  - A low gas price will mean the transaction is not processed quickly or maybe never
- If accepted, the miner processes until the gas limit is reached.
  - If a transaction fails or is incomplete, initiator still pay the fee (because resources were used), but the state of the EVM remains as it was before the attempt to process the transaction.

Why have Ethereum Gas?

- To ensure programmers pay for the cost of processing smart contracts
- To decouple payment for processing from the market value of the currency
- To eliminate infinite loops and hinder DoS attacks
  - Eventually an infinite program will run out of finite gas
  - Makes an attacker pay for the resources they use
- The more complex the programming commands requested, the more gas the initiator needs to offer.
- Facebook's proposed Libra cryptocurrency will also use gas
  - But the same coin will be used for gas and for the currency



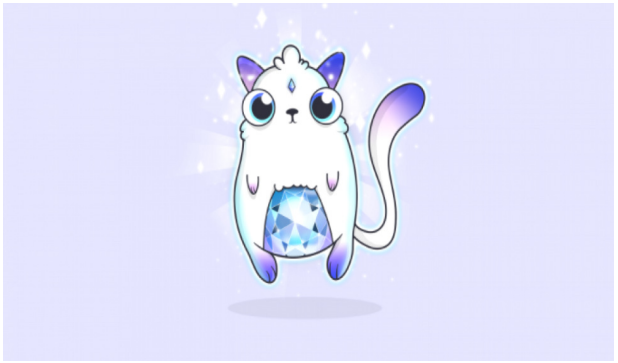
EVM = Ethereum Virtual Machine

- Ethereum Virtual Machine
  - Runtime environment for smart contracts
  - 256-bit register stack
- Isolated from the network and from the file systems of clients
- Implemented in
  - C++, Go, Haskell, Java, Javascript, Python, Ruby, Rust
- The platform is designed so that Smart Contracts, when processed by miners, will change the state of the EVM.
  - Hence, Ethereum has been called a global computer.

See:

Gavin Wood: "*Ethereum: A Secure Decentralised Generalised Transaction Ledger*". (EIP-150 Revision) Ethereum Yellow Paper.

Cryptokitties



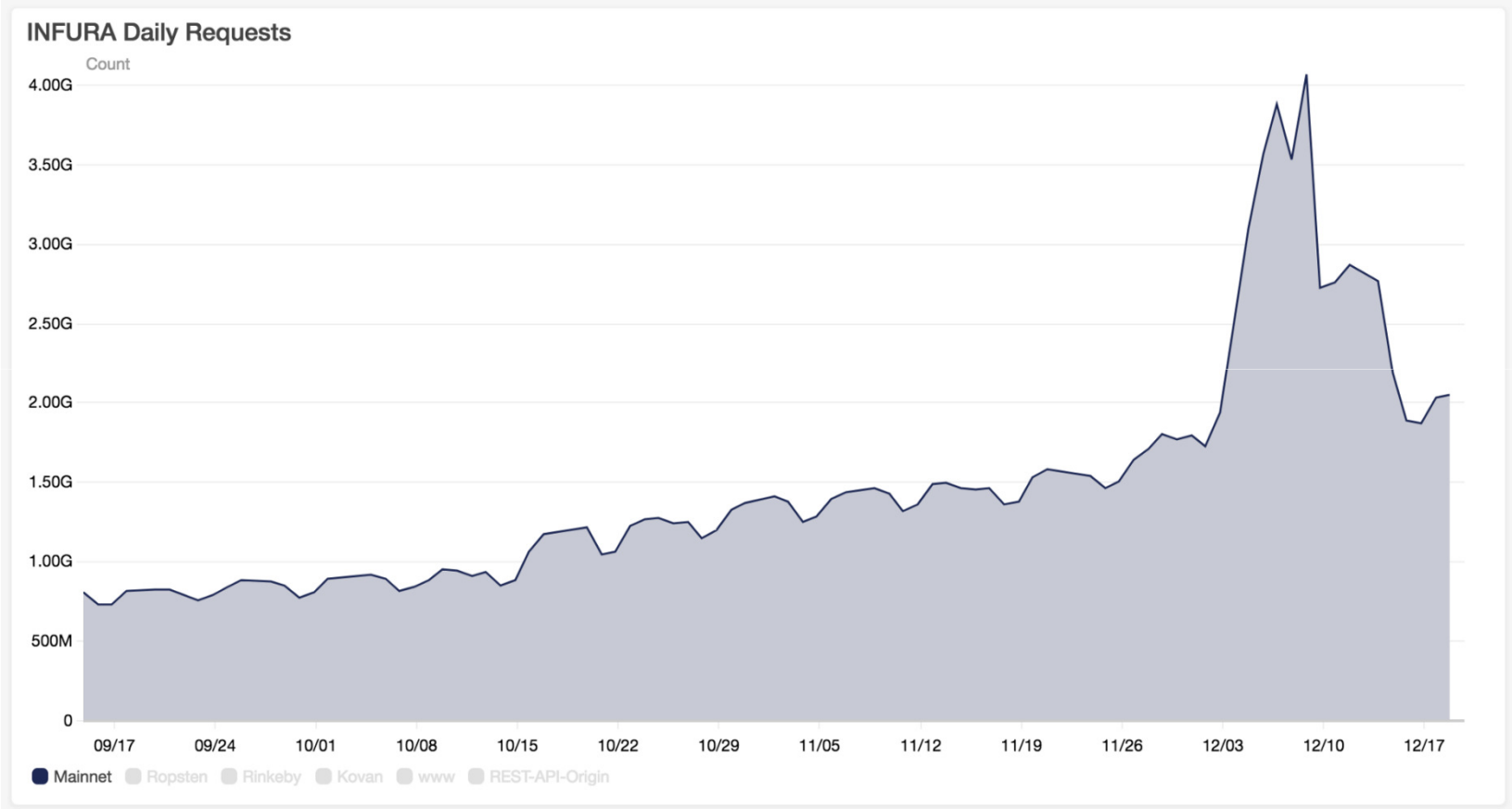
The Problem:

- Cryptokitties launched 12/2017
- Very popular
- Daily requests on Ethereum rose
  From 2 bn/day to 4 bn/day

- Usually such spikes in demand will resolve automatically
  - Miners accept higher gas prices
  - Transactions cost more & Market forces control the queue
  - This did not happen

- Response
  - Short-term: Modified front-end to allow users to resubmit their transaction with a higher gas price
  - Longer-term: Move dapps to sidechains.

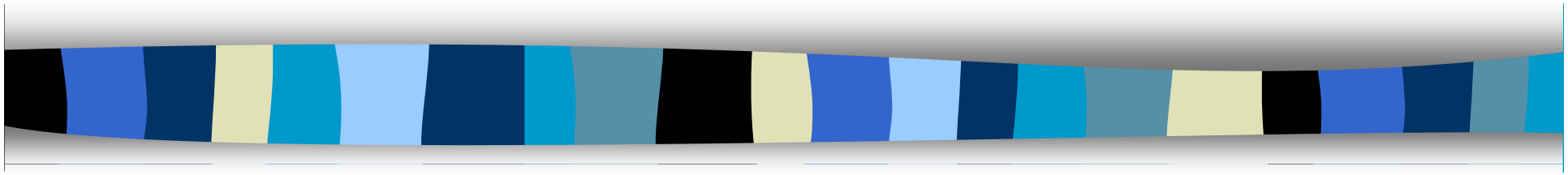
Daily requests on Ethereum through Infura nodes



Source: Consensys: *The Inside Story of the Cryptokitties Congestion Crisis.* Medium. 20 February 2017.

Ethereum Enterprise

- Ethereum MainNet is the open platform
- Permissioned chains may be run as sidechains
  - With various forms of synchronization to Ethereum MainNet
- Issues of how to manage permissions
  - Ethereum does not have a central node
- Software companies founded by Ethereum alumni:
  - Consensys (USA)
  - Parity (UK).



Corda

Corda

- Project initiated by a consortium of banks
  - R3cev LLC
  - HQ in New York, main work in London & Dublin
  - Consortium of 70+ banks and financial institutions
- Established 2015 to develop distributed ledger technologies for banking and financial applications
- Created platform called CORDA
  - Open-source DL platform released 30 November 2016
  - For scalability and support, will require enterprise version (which is licensed) called R3 Corda.

Problem

- Contracts between 2+ financial entities
  - Legal document
  - Shared facts are just between the entities
  - Visible to appropriate regulators
  - CAP theorem – different users may prioritise consistency over availability

Design

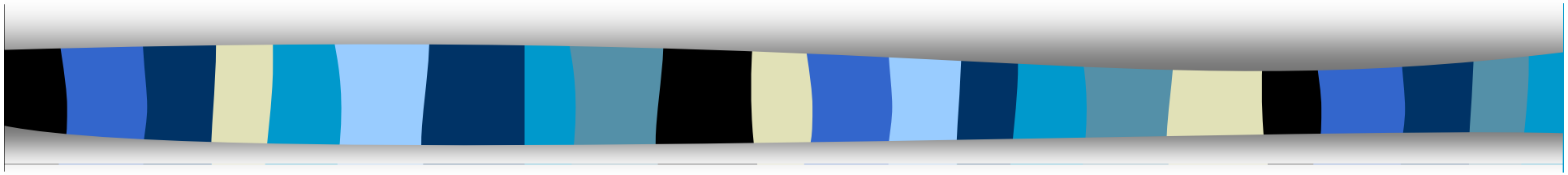
- Consensus
  - Just parties to a transaction, not the entire community
- Validation
  - Just legitimate stakeholders, not the entire community
- Use of Independent notaries
  - For time-stamping & prevention of replayed transactions
- Focus on inter-operability
  - With legal code
  - With legacy IT systems.

Corda features

- Transactions private to the parties involved
  - Witnessed by notaries
  - Only participants to a transaction can view it
  - Not even fact of transaction is seen by others
- No single chain
- No global state
- No native crypto-currency
- Records an explicit link between legal code and smart contracts
- Supports a variety of consensus mechanisms
- Can include transaction within arbitrary workflows.

Enterprise version: R3 Corda

- R3 Corda is an enterprise version of Corda
- Differs from open-source version in terms of
  - Support & maintenance
  - Node capabilities
  - High availability & disaster recovery
  - Management & monitoring
  - Enterprise network configuration/ firewalls
- Access to template libraries of code and workflows
  - Eg, dispute resolution workflows
  - Flow Hospital – admin can see flows which need fixing
- Integration to other protocols
  - Corda interfaces to RPC (Remote Procedure Call) prototocl
  - R3 Corda interfaces to FPML (Financial Products ML) / SWIFT ISO-20022
- Privacy & Key Management, eg
  - Role-Based Access Control in R3 Corda, not in Corda
  - Light-weight Directory Access Protocol (LDAP)
- Pluggable crypto
- Ability to operate Corda nodes inside a corporate firewall
  - Enterprises usually averse to P2P networking
- Operational GUI.



Hyperledger

Hyperledger

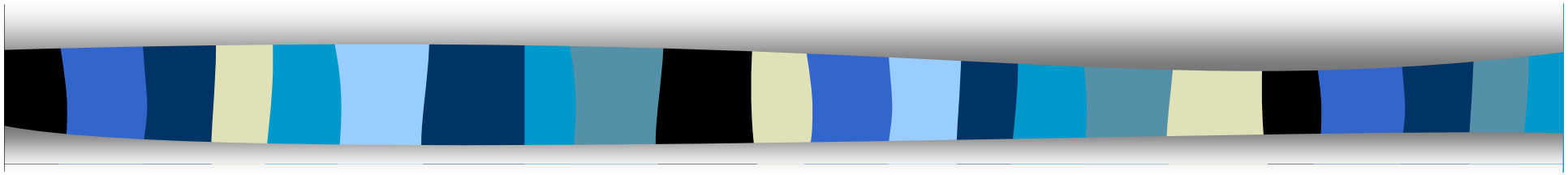
- December 2015 – Started by Linux Foundation
- IBM contributed code from OpenBlockchain
  - Became Hyperledger Fabric
- Early members:
  - Tech firms - IBM, Intel
  - Financial Institutions - ABN AMRO, JP Morgan
  - Software companies – SAP
  - Systems integrators – Accenture, Wipro
- Umbrella project for related projects.

HYPERLEDGER

Platforms

HYPERLEDGER

- HL Burrow
  - an Ethereum Virtual Machine via HL
  - Monax & Intel
  - Discussed last week by Luke Riley
- HL Fabric
  - IBM
  - Permissioned BL infrastructure
  - Chaincode (smart contracts)
  - Configurable consensus & membership services
- HL Sawtooth
  - Intel
  - Proof of Elapsed Time (PoET)
  - Hardware-based security
- HL Development tools.



Other technologies

Other technologies

- Common criticism - You don't need a DL platform, you could do this with a centralized database and private messages

- Technically correct. But –
  - Who holds this database? How is it paid for?
  - Who can be trusted not to exploit it?
  - Risk of attack?
  - Regulatory issues (eg, anti-trust laws)

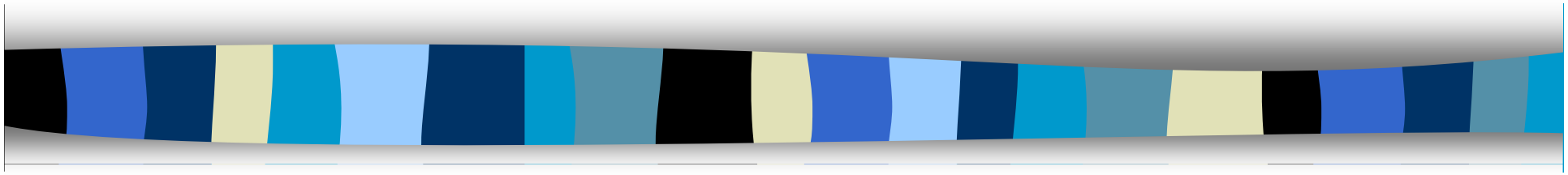
- Anti-trust (pro-competition laws)
  - Preclude any member of a partnership holding commercial data from competitors
  - So: A centralized database needs to be held by a third-party host

- Commercial considerations
  - A third-party host is more expensive
  - A third-party host may be able to monetize the shared data.

Enhanced database technology

A central database with secure private messaging and with

- A private network (closed to entities without permission)
- Digital signatures (public/private keys) for secure identity
- Encryption of messages between participants
- Each participants holding relevant data in their own database
- A consensus mechanism (possibly)
- To ensure immutatibility, periodic hashing of database contents to a public blockchain (eg Ethereum)
  - Eg, Everledger in diamonds.



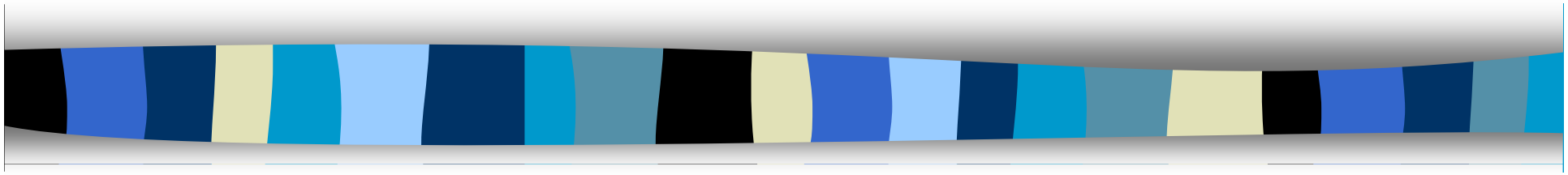
Comparison of Platforms

	Ethereum	Corda	Hyperledger	P2P & Hashes
Technical				
True DL?	Yes	Yes	Yes	No
How is state of the system decided?	Validation by all system participants, in accordance with consensus mechanism.	Validation by notaries and parties to each transaction, in accordance with consensus mechanism,.	Validation by all system participants, in accordance with consensus mechanism.	No central system state. Participants agree transactions and then an upload to a blockchain establishes proof-of-existence at time of upload.
Consensus Mechanism	Proof-of-Work. Moving to Proof-of-Stake in 2018. In the transition period, both PoW & PoS will be used.	Validation by participants to a transaction & by independent notaries. Could work with any consensus model, including ones supported by SGX (Intel enclave technology)	Modular structure that could work with any consensus model. May use Proof-of-Authority or Voting-based	N/A. Voting protocols would need to be developed,
Smart contract capabilities	Yes Solidity language (still immature)	Yes In Java and Kotlin	Yes Most mature in Go language. Java is also possible, but still immature.	No Would need to be developed.
Development history	Open source	R3 CEV	IBM and others	The ideas arise from the world of distributed databases.
Development outlook	Good	Good	Good	N/A
Limitations & Flexibility	Public/open	Focused on financial applications	Modular structure makes this flexible Privacy possible but weak and non trivial design.	Need to ensure flexibility and scalability in designing the P2P communications

Scalability	Good With mainnet Ethereum scalability could be seen as extensive.	Medium	Medium	Unclear
Transaction performance	1 Transaction per 15 sec With Plasma and the Lightning network which introduces parallel child blockchains this can improved drastically, to Billions transactions per second.			
Security & Identification	Public/open or Private/closed when set up as a permissioned network.	Private	Private	Private
Non-Technical				
Key backers/ investors	Ethereum Foundation Open-source community Support from many later ICOs.	R3 CEV Some 70+ global banks	IBM Intel Linux Foundation and over 100 others	Database vendors
Vision by	Vitalik Buterin	Mike Hearn	IBM ?	?
Usability	Needs to be developed	Needs to be developed	Needs to be developed	Needs to be developed
Support community	Broad	Financial community	Limited	?
Standardization & adoption levels	Widespread	Support in banking	Unclear	N/A

Conclusion: It is still early days

- The technology is still immature & functionalities are limited
  - Eg, Scalability is still an unknown
- Dev Tools are still immature
- Development experience is limited
  - Bitcoin Blockchain and Ethereum are still the only large-scale deployment of DLT technology
  - There are, as yet, still no large-scale commercial applications
  - Systems in production: Vakt / Komgo / InsurWave
- Consortium of energy trading companies & banks recently a trial
  - Open Development Challenge (ODC)
  - December 2017-January 2018
  - Approx. 50 companies approached, 10 invited to build PoC (2 weeks)
  - Different platforms trialled
  - No platform is obviously or uniformly best for this problem domain.



Case Study

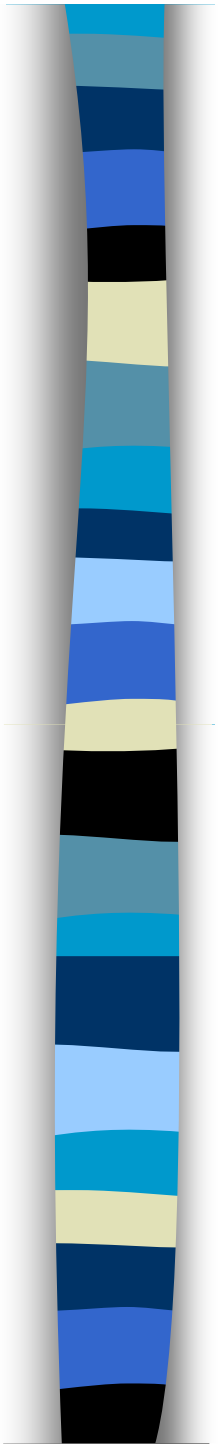
Case Study: Post-trade energy commodities

Deal: Energy trading company A agrees with B to deliver 1 million barrels of Brent Crude sweet light oil to Energy trading company B in Amsterdam on 1 May 2018 for $65 per barrel (total price $65 million).

- B needs to commission empty oil tankers to Amsterdam by 1 May 2018
- B (or shipping company) needs to book slots in Amsterdam port for these tankers
- B needs to commission Inspectors to inspect the oil prior to its offloading from the tankers of A
- B needs to have $65 million ready to pay A on 1 May 2018
- B needs to take out insurance for the oil upon receipt
- A needs to make sure it has 1 million barrels of to Brent Crude to give to B in Amsterdam on 1 May 2018.
- A may take out insurance on its oil
- A may require bridging finance between now and when the money from B is delivered.

Problem definition

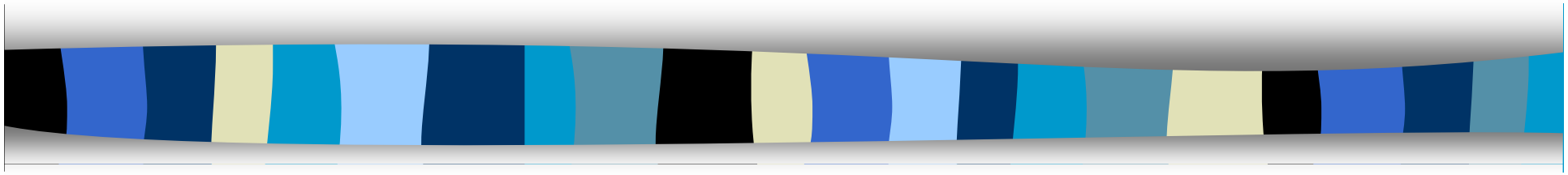
- Current technology
  - Comms via phone & email
  - Sharing of contracts in PDF docs
  - Some legal constraints
    - Eg, Bills of Lading (BoLs) have to be in paper form in some countries
- Problem:
  - Each company has its own database
  - Same data entered often, by different companies
  - Need for data reconciliation
  - Not automatable
  - Lots of double-spend problems
    - Assignment of oil to ships, ships to ports, etc (“nominations”)
- Desire:
  - Data only entered once (shared values)
  - Data shared but only between participants
  - Regulators need appropriate access
  - Management of workflows & chains.



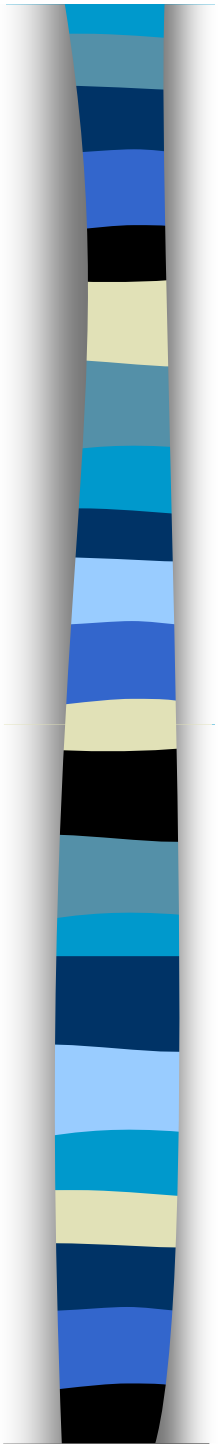
Design considerations

- Desire some form of shared ledger
- Permissioned vs Permission-less?
- Should there be a central node?
  - For authorizations
  - Granting identity keys
  - Software version management
- How to ensure
  - Private blockchain is run efficiently?
  - Participants can assign access rights to data to downstream participants?
- How to avoid
  - Participants seeing data that they should not?
  - Central node accessing or monetizing data?

Thank you!



peter.mcburney@kcl.ac.uk



Exercises

1. Articulate a system design for the Case Study problem.

2. Consider Facebook's proposed Libra cryptocurrency. Starting with the Libra Whitepaper, explain the technical architecture of the Libra Blockchain:

   https://libra.org/en-US/white-paper/