A Breathless Tour of Blockchain

Eoin Woods
Endava

Nick Rozanski
ICBC Standard

Agenda

- What is Blockchain?
- Blockchain as an Architectural Element
- Applications for Blockchain
- Programming the Blockchain
- Summary

What is Blockchain?

What is Blockchain?

- The enabling technology of Bitcoin
- A distributed database without a controlling authority
- An auditable database with provable lineage
- A way to collaborate with principals you do not trust
- Trust and storage mechanism for cryptocurrencies
- A possible architectural component for highly distributed and reliable Internet-scale systems

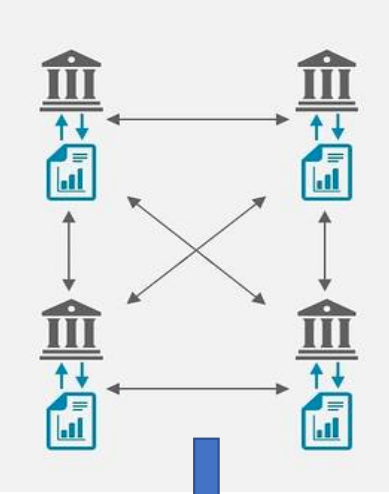
What is Blockchain? Some Terms

bitcoin

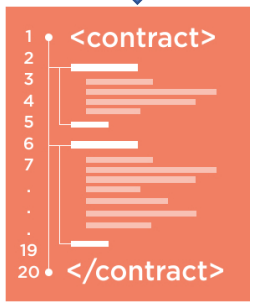
cryptocurrency implemented using a blockchain

BLOCKCHAIN

cryptographic Implementation of a distributed ledger

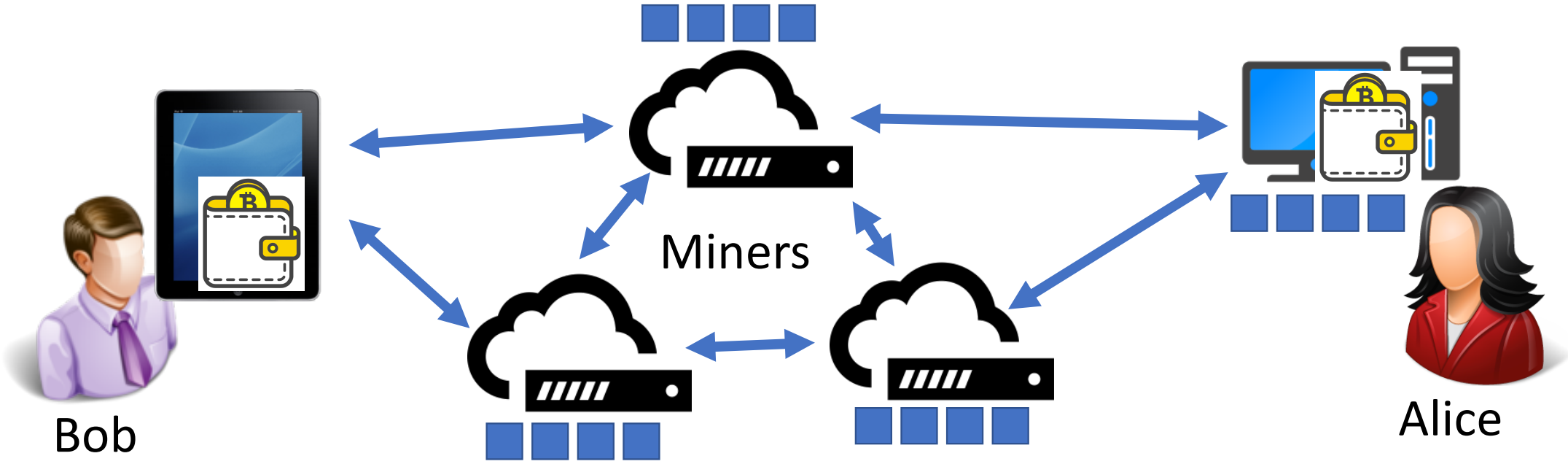


<u>Distributed Ledger</u>
fully replicated append-only shared data store which achieves trust without a central authority

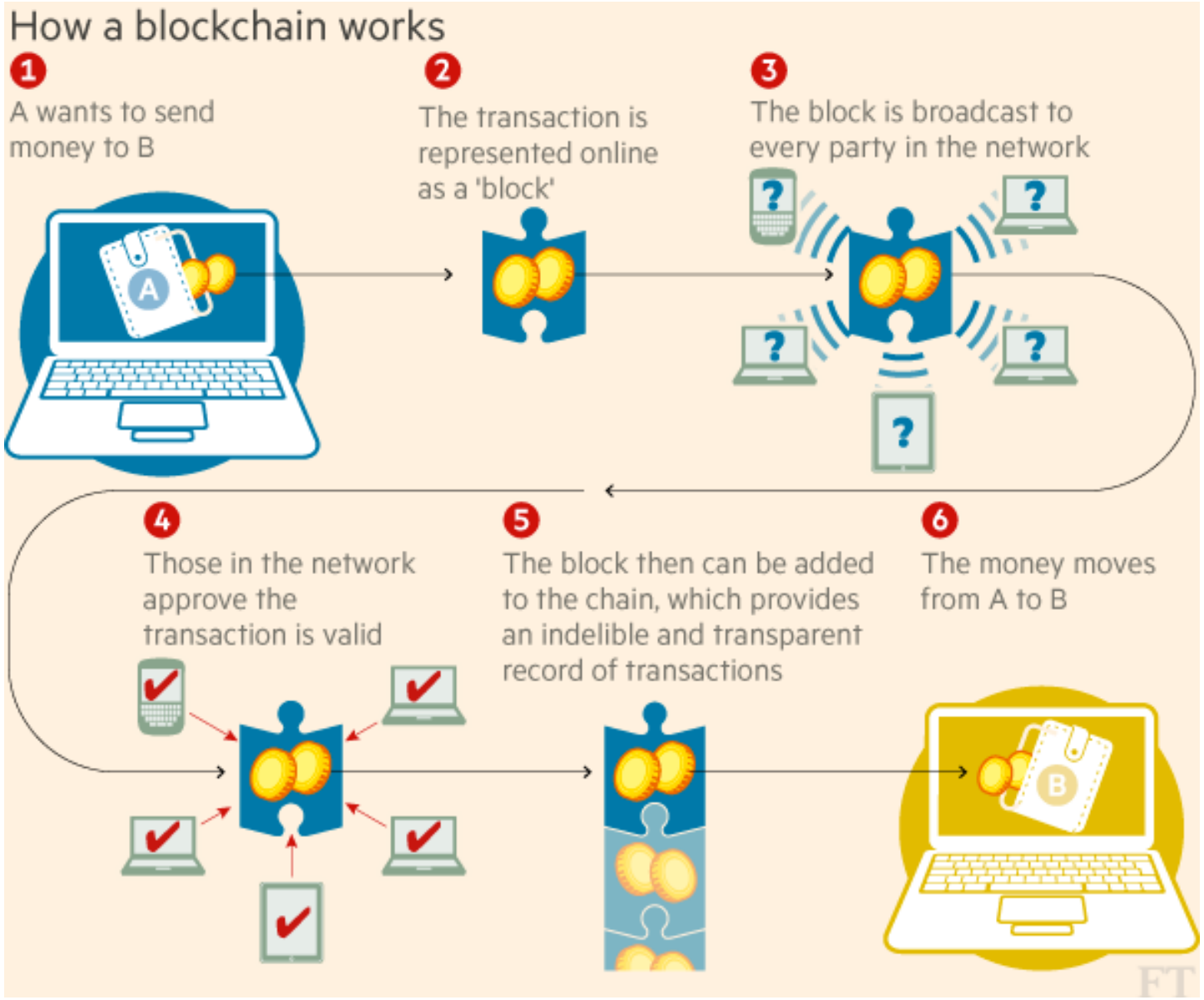


<u>Smart Contract</u>
code stored in the ledger executed when preconditions met to manipulate ledger state

What is Blockchain? Bitcoin network example

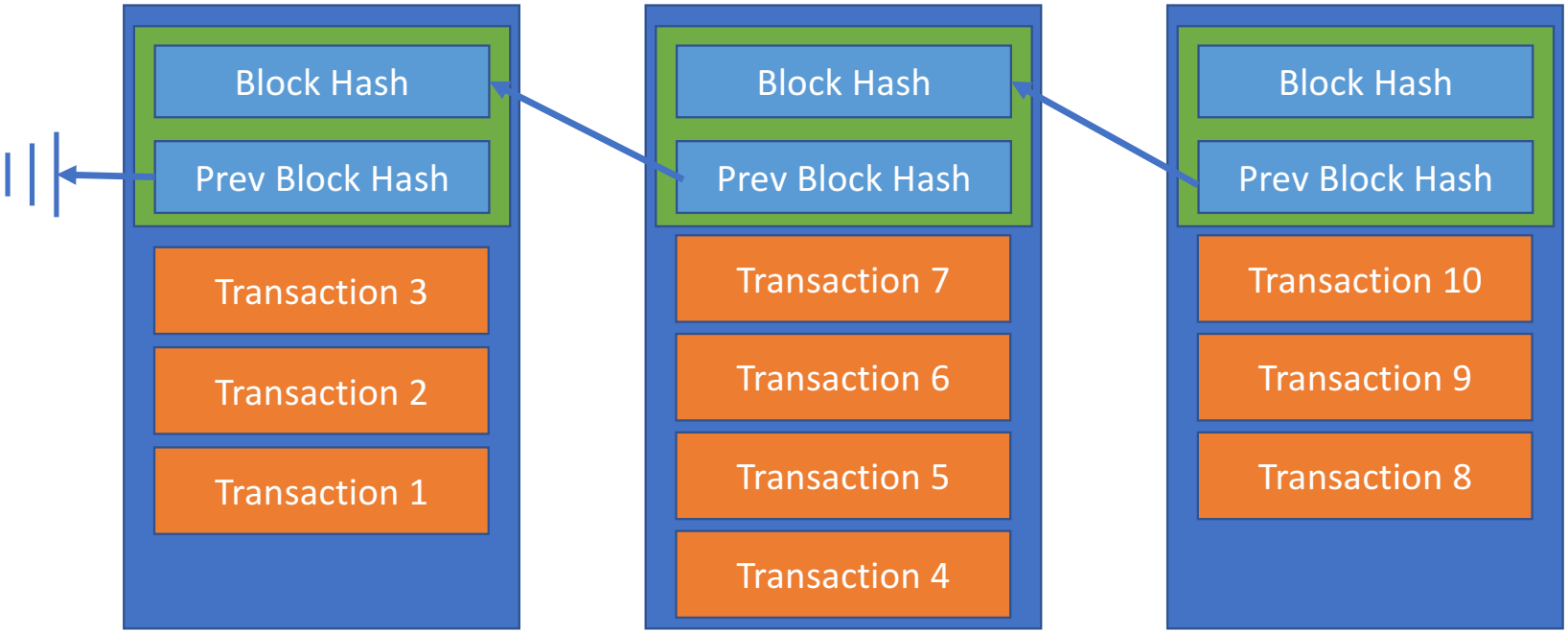


Blockchain Operation



Banks seek the key to blockchain - Financial Times

What is Blockchain? Blockchain structure



- Public keys used to identify participants
- Bitcoin addresses identify participants in a transaction (derived from PK)
- Cryptographic hashes used to ensure integrity

What is Blockchain? Consensus

Q. How do we know the information on the chain is correct?

A. Because everyone agrees – this is “consensus”

Blockchains use different types but “proof of work” is common

- To create (“mine”) a block you need to solve a hard problem
- If you don’t solve the problem then peers will reject your block
- Thus forging the blockchain would require a huge amount of work to get your fraudulent blocks accepted (“impossible” without 51% of capacity)

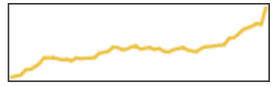
Other common models are “proof of stake” and “proof of membership”

What is Blockchain? What blockchains exist?

A lot of blockchain implementations exist. A small sample of them are:

Bitcoin	2009	Cryptocurrency
Litecoin	2011	Cryptocurrency
Ripple	2012	Blockchain payment and settlement system
Chain	2014	Enterprise blockchain
Ethereum	2015	Blockchain dapp platform
Hyperledger	2015	Linux Foundation open source blockchain projects
R3 Corda	2016	Distributed ledger for the financial industry
Multichain	2017	Enterprise blockchain

Cryptocurrencies

#	Name	Market Cap	Price	Circulating Supply	Volume (24h)	% Change (24h)	Price Graph (7d)
1	Bitcoin	$47,495,108,171	$2898.94	16,383,612 BTC	$1,802,610,000	2.06%	
2	Ethereum	$29,528,845,801	$319.68	92,368,857 ETH	$1,140,100,000	16.64%	
3	Ripple	$10,237,398,031	$0.267111	38,326,381,283 XRP *	$140,686,000	-7.89%	
4	NEM	$1,836,081,000	$0.204009	8,999,999,999 XEM *	$19,892,000	-6.91%	
5	Ethereum Classic	$1,684,219,283	$18.23	92,388,248 ETC	$135,953,000	4.54%	
6	Litecoin	$1,550,596,067	$30.12	51,485,057 LTC	$182,244,000	1.07%	
7	Dash	$1,103,268,428	$150.01	7,354,387 DASH	$28,747,500	0.95%	

source: coinmarketcap.com

7 cryptocurrencies have a market cap of > $1B! Bitcoin market cap is $47B.

What is Blockchain being Used For?

verifiable supply chains

everledger

digital ledger that tracks and protects valuable assets

IBM

MÆRSK

supply chain efficiency

epiphyte

Trade Settlement and FX Transactions

Digital Asset Holdings

ASX

post-trade processing

Keybase

Identity management

TIERION

verified data

BitFury

Georgia government records

Blockchain in Finance

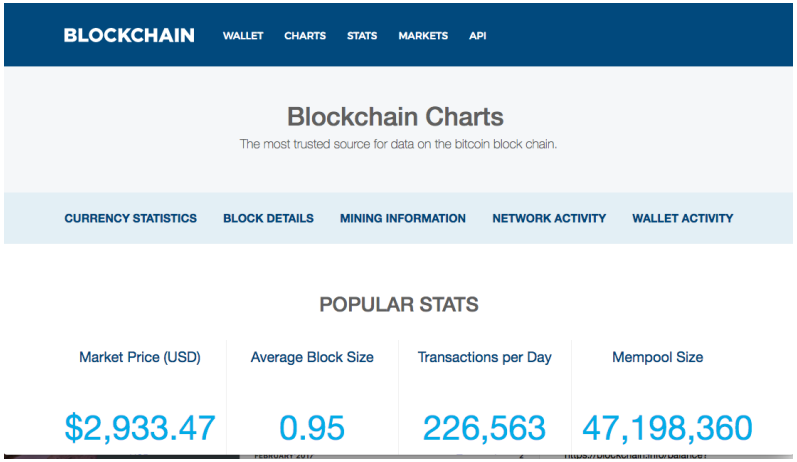
Huge interest from institutions, utilities and regulators

github.com/eoinwoods/spacoin

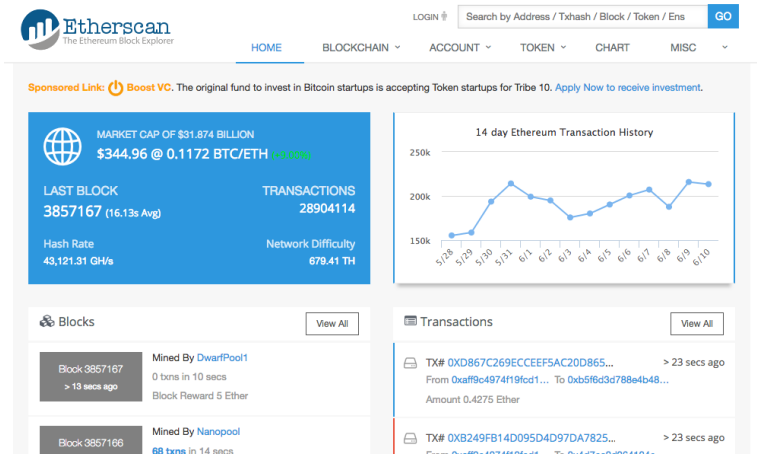
Exercise: Browse a blockchain

Most public blockchains have a range of associated web based “explorers”

- For Bitcoin we’ll use blockchain.info
- For Ethereum we’ll use etherscan.io



blockchain.info



etherscan.io

Exercise: Explore a blockchain

5 minute exercise to look at blockchain content

- The WannaCry ransomware attackers have 3 bitcoin wallets:
  - 13AM4VW2dhxYgXeQepoHkHSQuy6NgaEb94
  - 115p7UMMngoj1pMvkpHijcRdfJNXj6LrLn
  - 12t9YDPgwueZ9NyMgw519p7AA8isjr6SMw
- Use blockchain.info to find out how much they have collected in ransom so far
  - Hint: there are 10-8 satoshis in a bitcoin and blockchain.info returns most values in satoshis, so multiply by 0.00000001 to get BTC)

Blockchain as an Architectural Element

Characteristics of a Blockchain Ledger

- Distributed, replicated database
- Distribution via a p2p model (no master)
- Consensus model used to ensure integrity
- Append only immutable store
- Highly fault tolerant
- Eventually consistent

Adding Smart Contracts

- Smart contracts are code which is stored on a blockchain and executed by the blockchain virtual machine
- Smart contract languages vary from very primitive (Bitcoin) to Turing-complete languages (Go on IBM Hyperledger, Solidity on Ethereum)
- Programming and execution models vary by platform
- We're going to use Ethereum Solidity as our example in this session

Example: Ethereum Solidity Smart Contract

```
contract Coin {
        // The keyword "public" makes those variables readable from outside.
        address public minter;
        mapping (address => uint) public balances;
        // Events allow light clients to react on changes efficiently.
        event Sent(address from, address to, uint amount);
        // This is the constructor whose code is run only when the contract is created.
        function Coin() { minter = msg.sender; }
        function mint(address receiver, uint amount) {
                if (msg.sender != minter) return;
                balances[receiver] += amount;
        }
        function send(address receiver, uint amount) {
                if (balances[msg.sender] < amount) return;
                balances[msg.sender] -= amount;
                balances[receiver] += amount;
                Sent(msg.sender, receiver, amount);
        }
}
```

We'll get to the details a little later …

Quality Properties of a Distributed Ledger

Positive Qualities

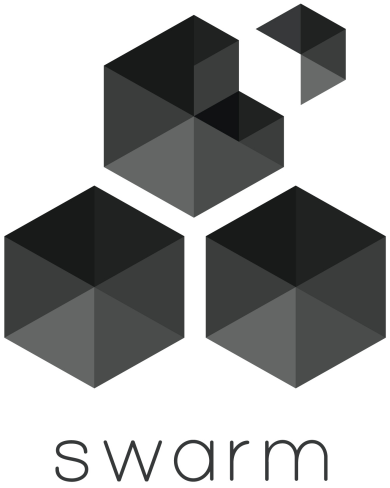
- Immutability once written
- Security (integrity, non-repudiation, availability)
- High fault-tolerance
- No single control or trust point
- Embedded immutable logic
- Dynamic, evolving

Negative Qualities

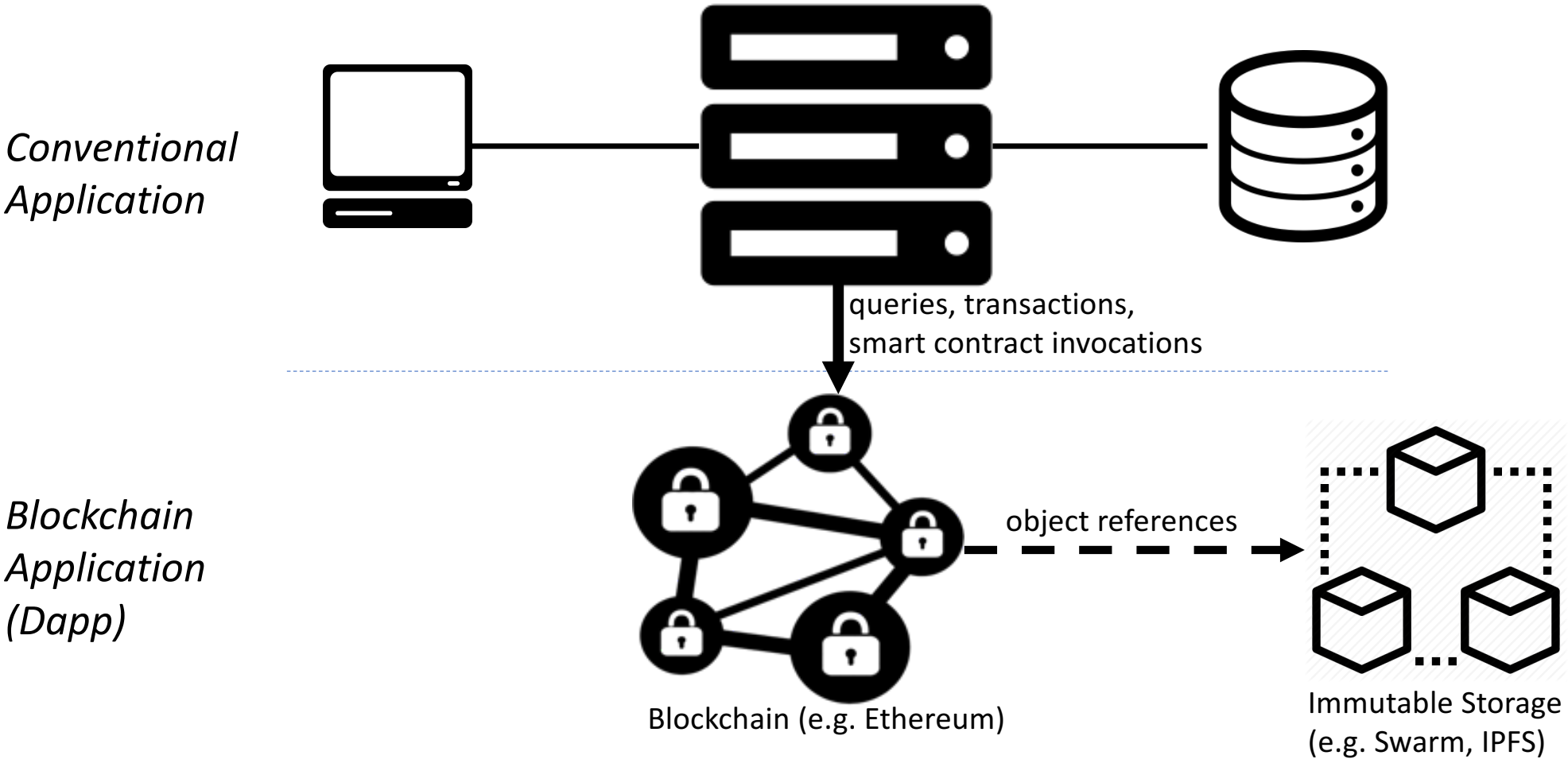
- (very) eventual consistency
- computationally expensive (generally)
- Limited query model
- Lack of privacy
- lack of throughput scalability (generally – 10s txn/sec)
- Lacking maturity

Aside: Storing Data

- Blockchains don’t necessarily store large amounts of data well
- Some (Bitcoin) optimised for many small transactions
- Solution is to store large data items in immutable storage and reference from blockchain
- IPFS and Swarm best known



Integrating Blockchain



Exercise: Applying Blockchain

- What could you use blockchain for?
  - Could be at work or an idea to change the world!
- What problems would it solve or introduce?

<u>Recap</u>

- *distributed, highly reliable, auditable, immutable database, not requiring trust between participants*
- *smart contracts can embed computation in it*
- *but slow, eventually consistent, limited queries*

Programming Blockchain

Smart Contracts

- The “programming” bit of blockchain technology
- Code that is held in the blockchain and executed by the runtime environment when certain conditions become true (e.g. a transaction occurs)
- Any participant can add a smart contract to a blockchain (for a fee)
- A smart contract usually mutates the state of the blockchain (e.g. by adding another transaction to it)
- Transforms blockchain from a passive database to a dynamic system
  - A bit like triggers and stored procedures in RDBMS

Bitcoin “Script”

- Very simple “FORTH-like” virtual machine
- Constrained programming model of “locking” and ”unlocking” scripts
  - Locking script (“scriptPubKey”) defines constraints to execute the transaction
  - Unlocking script (“scriptSig”) satisfies the constraints to allow execution
- Small number of “op codes” for use in scripts.

<u>Example</u>

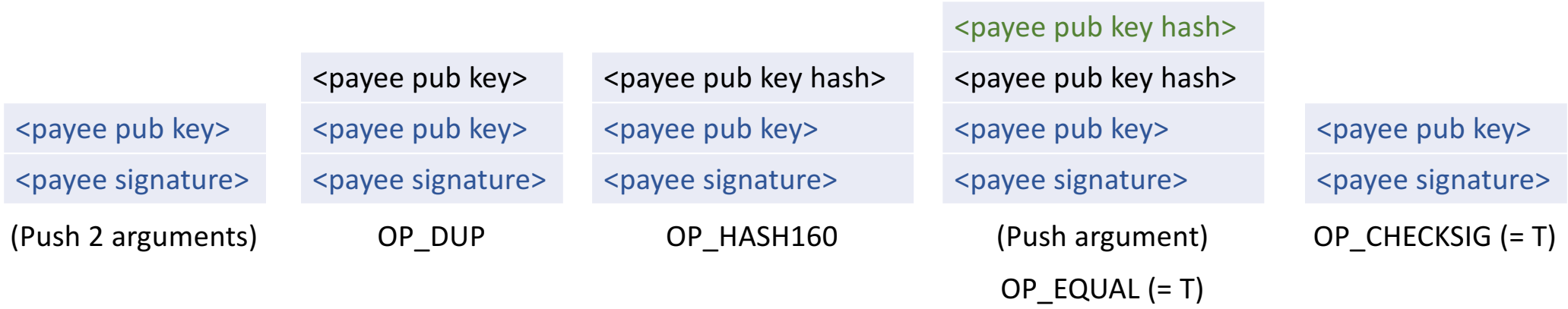
Lock: `OP_DUP OP_HASH160 <payee pub key hash> OP_EQUAL OP_CHECKSIG`

Unlock: `<payee signature> <payee pub key>`

Bitcoin “Script”

```
<payee signature> <payee pub key>
OP_DUP OP_HASH160 <payee pub key hash> OP_EQUAL OP_CHECKSIG
```

Stack based execution:



Ethereum Solidity

- Most popular contract language for Ethereum
- Syntax quite similar to JavaScript (but statically typed language)
- Turing complete, object-oriented language
- Inheritance and user-defined types
- Compiles to bytecode that runs on the EVM
- Emerging development eco-system of frameworks and tools

Ethereum Solidity

```
contract Coin {
        // The keyword "public" makes those variables readable from outside.
        address public minter;
        mapping (address => uint) public balances;
        // Events allow light clients to react on changes efficiently.
        event Sent(address from, address to, uint amount);
        // This is the constructor whose code is run only when the contract is created.
        function Coin() { minter = msg.sender; }
        function mint(address receiver, uint amount) {
                if (msg.sender != minter) return;
                balances[receiver] += amount;
        }
        function send(address receiver, uint amount) {
                if (balances[msg.sender] < amount) return;
                balances[msg.sender] -= amount;
                balances[receiver] += amount;
                Sent(msg.sender, receiver, amount);
        }
}
```

Contract

Typed state

Event for log (and callback)

Functions to operate on state

Solidity & Ethereum Development Ecosystem


Geth


Eth


Pyethapp


Mist


Netereum

Web3.js


Web3J

API Libraries

testrpc

Nodes & Browsers

Dapp

Embark


Solidity Browser


Ethereum Studio

Development Tools

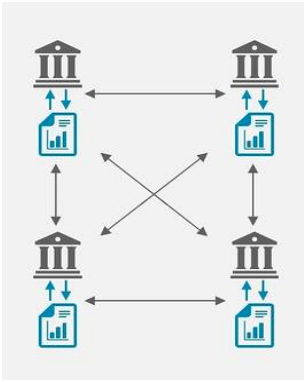
Demo: Programming Blockchain

- Solidity code and the “Sol” compiler
- Writing and unit testing a simple Solidity contract using “Truffle”
- Deploying a compiled contract to a local test network

Summary

Key Concepts

Distributed Ledgers



Blockchains

Smart Contracts

What is Blockchain?

- A novel type of **distributed database**
- Append-only, auditable, fault tolerant, secure **by design**
- Slow and high-latency by accident
- Many blockchains (and distributed ledgers) can host "**smart contract**" code to provide secure computations

Some Applications

- **Value transfers** – securities clearing, payments
- **Verifiable records** – property registers, supply chains, loyalty points
- **Identity** – verifiable personal or corporate identity, passports
- **Verifiable Storage** – immutable, secure storage and sharing
- **Decentralised Notary** – prove existence of digital asset at time
- **Currencies** – Bitcoin, Ether, Litecoin and friends

To Find Out more

- **Bitcoin** – bitcoin.org
- **Ethereum** – ethereum.org
  - Solidity - solidity.readthedocs.io
  - Truffle - truffleframework.com
  - Embark - github.com/iurimatias/embark-framework
  - Dapp - dapp.readthedocs.io
- **IPFS** – ipfs.io
- **Swarm** - swarm-gateways.net
- **News** – coindesk.com, cryptoinsider.com (and many others)

Thank You

Eoin Woods
Endava
eoin.woods@endava.com

Nick Rozanski
ICBC Standard
nick@rozanski.org