Blockchain

The Hitchhiker's Guide

pwc

www.pwc.com/digital

Introductions

Michael McFall

Solution Architect / Principal Data Engineer

michael.mcfall@uk.pwc.com

- 14 years in the industry - specialising in data
- Roles including *developer, DBA, data engineer & architect*
- Last 3 years spent exploring, designing & building blockchain platforms
- Currently with **PwC's Blockchain delivery group** in Belfast (NI)

First, a question...

How much do you know about Blockchain already?

What is a Blockchain?

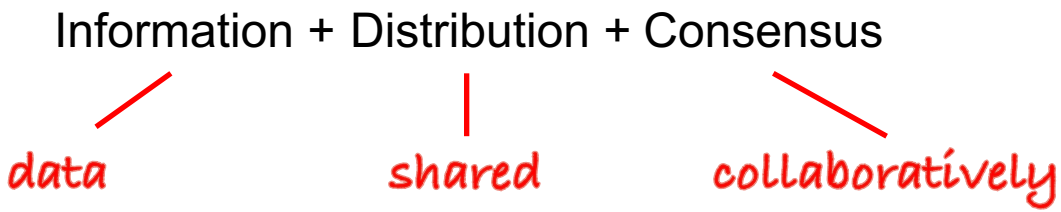
“A blockchain is a data structure that makes it possible to create a digital ledger of transactions and share it among a distributed network of computers. It uses cryptography to allow each participant on the network to manipulate the ledger in a secure way without the need for a central authority…

WALL STREET JOURNAL – *CIO Journal*

Ummm…

Lets try again…

A blockchain allows data to be safely shared between many participants…



Information

Think of it as a ledger

Distribution

…where everyone has a complete, up-to-date copy

Consensus

…and where no-one can make changes without agreement from others


How do they work?

Building Blocks

Blockchains are underpinned by three main concepts…

Shared View

of data for all participants in a peer-to-peer (p2p) network

Consensus

via algorithms used to reach agreement on the state of the system

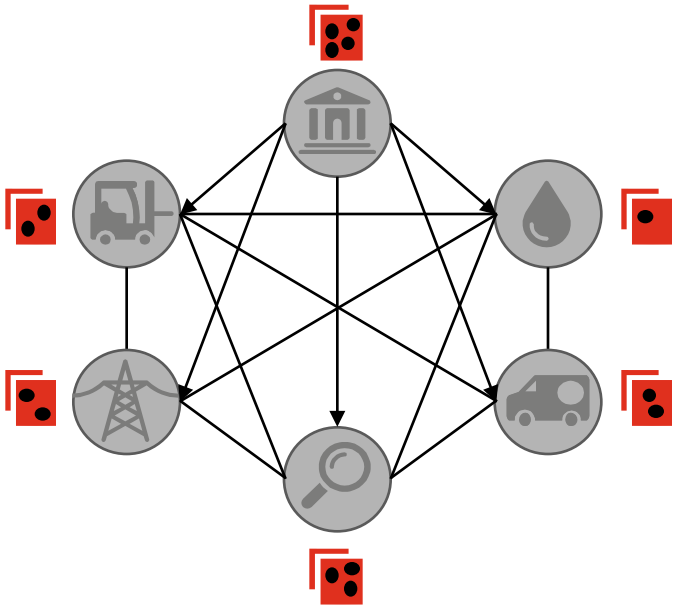
Cryptography

used to establish identity and protect the integrity of the underlying data

Shared View

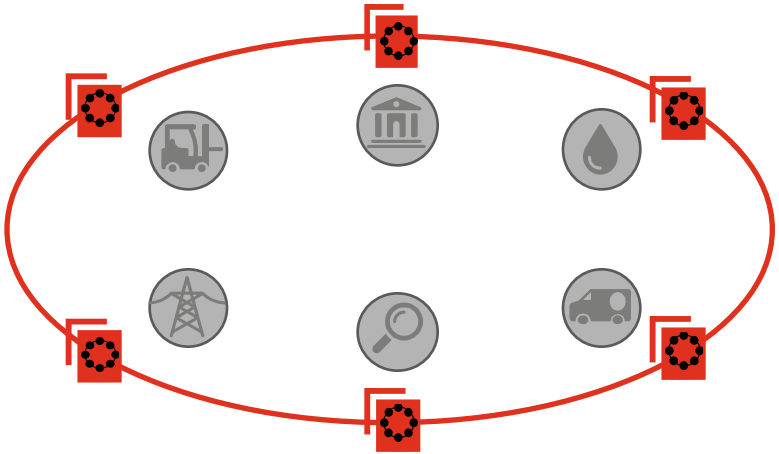
Traditional Approach

Each party maintains their own ***independent*** ledger



Blockchain Approach

All parties share and maintain the ***same*** ledger



Consensus

The means by which the participants in the blockchain can ***reliably*** and ***methodically*** come to agreement on the state of the system.

A number of different consensus schemes exist: **proof-of-work** (PoW) and **proof-of-stake** (PoS) are commonly used examples on Blockchains.

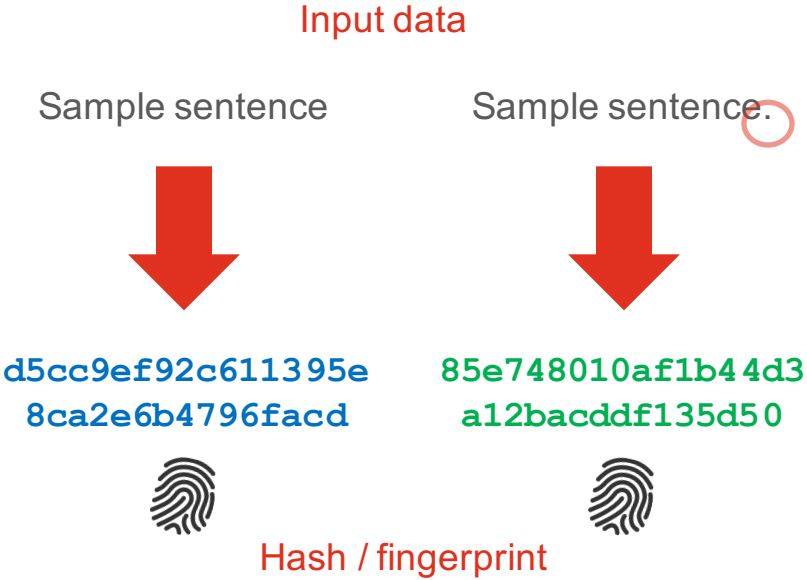
Cryptography - ***Hashing***

Generates a ***fingerprint*** for a piece of data by applying a cryptographic function to it.

Changing one character in the original string results in a completely different hash.

The original string cannot be reverse engineered from the hash.

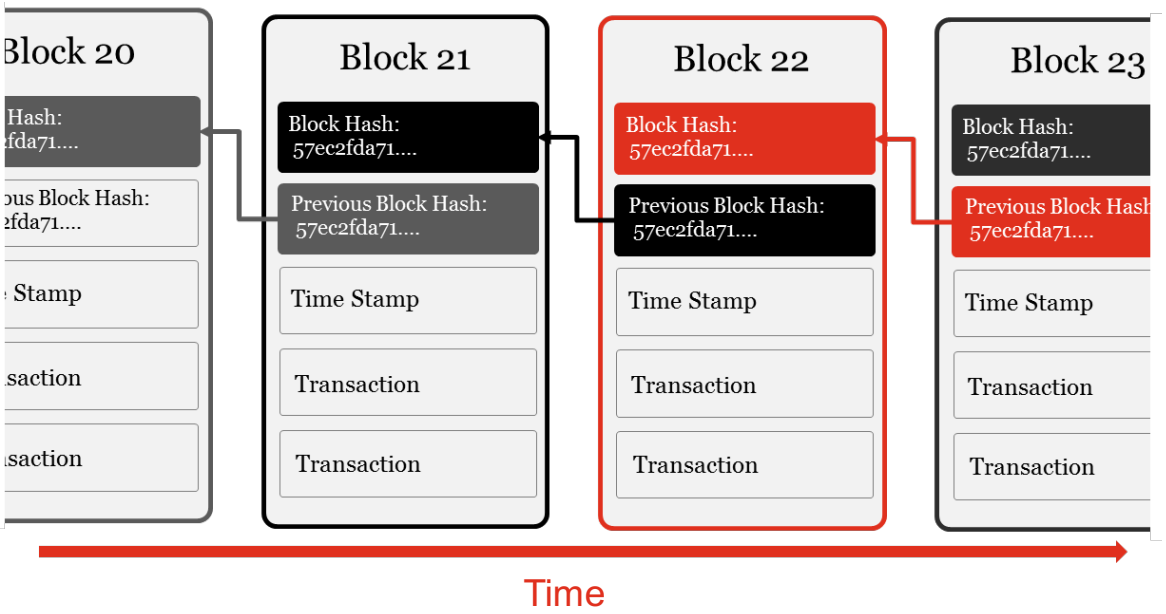
Hashing is an effective means of determining if a piece of data has been changed.



Cryptography – ***Chain of Blocks***

Each block is cryptographically hashed to generate a '*fingerprint' (block hash)*. This hash is included in the next block, which creates a chain that can be traversed from the latest block, all the way back to the 1st block ever generated (the *genesis block*).

Any attempt to change the content of a block will 'break the chain' – this provides a resilient and tamperproof record of events.



Cryptography – ***Public & Private Keys***

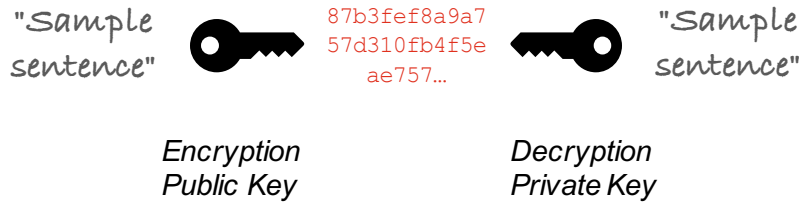
Keys are generated from large prime numbers that are mathematically related. Data encrypted with one key can only be decrypted with the other, and vice-versa.

Encryption scrambles data with the public key, so the holder of the private key (i.e. the recipient) is the only one able to decrypt the message.

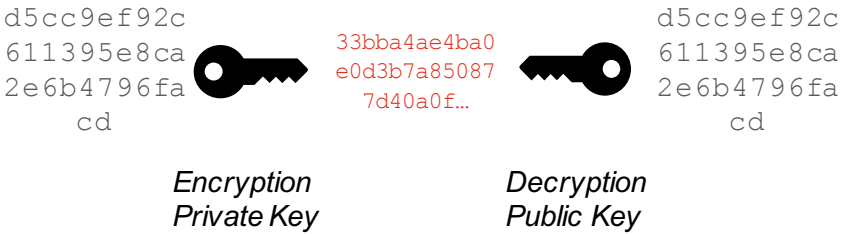
Digital Signatures encrypt data with the private key of the sender, and can be decrypted & verified by anyone using the public key.

Public key cryptography forms the basis of identity verification and secure ownership in a blockchain

Encryption



Digital signature



Smart contracts

Some Blockchains also provide the ability to perform specific computation on the chain, through the use of ***smart contracts***.

Smart contracts can be encoded to apply predefined business logic to govern **what** can be performed in the contract, **when** it should occur, **how** it must be executed, and by **who**.

```
contract token {
  mapping (address => uint) public coinBalanceOf;
  event CoinTransfer(address sender, address receiver, uint amount);

  /* Contract initialised with initial supply tokens to the creator */
  function token(uint supply) {
     coinBalanceOf[msg.sender] = supply;
  }

  /* Simple trade function */
  function sendCoin(address receiver, uint amount) returns(bool sufficient){
     if (coinBalanceOf[msg.sender] < amount) return false;
     coinBalanceOf[msg.sender] -= amount;
     coinBalanceOf[receiver] += amount;
     CoinTransfer(msg.sender, receiver, amount);
     return true;
  }
}
```

How are they deployed?

Permissionless

Anyone is free to participate. Transactions are public, while identities are not always known

Permissioned (public)

Open network where identities are known and participant roles are agreed

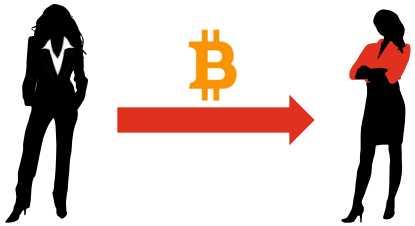
Permissioned (private)

Fully *private* or *consortium* networks, where visibility and participation is restricted only to authorised entities

Example - ***Bitcoin***

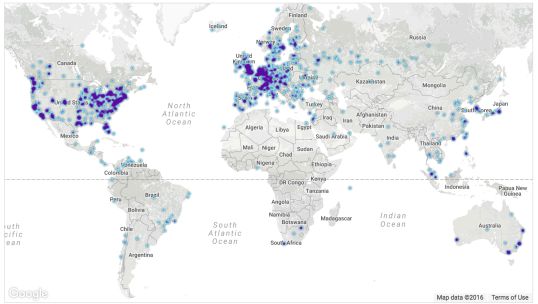
Purpose

A global payment network for the Bitcoin cryptocurrency



Topology

A public global network[1] of **~6,000** nodes



Features

- ✅ Censorship-resistance
- ✅ Cross-border TXNs
- ✅ No chargebacks
- ✅ Push payments = low fraud
- ✅ Fast & inexpensive*
- ✅ Ability to extend via *coloured coins* & *sidechains*

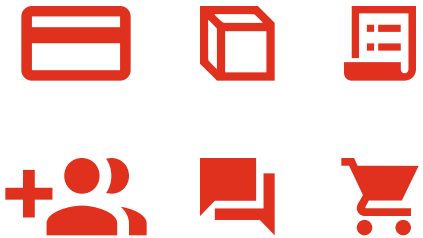
[1] **Bitnodes** - https://bitnodes.21.co/

Example - ***Ethereum***

ethereum

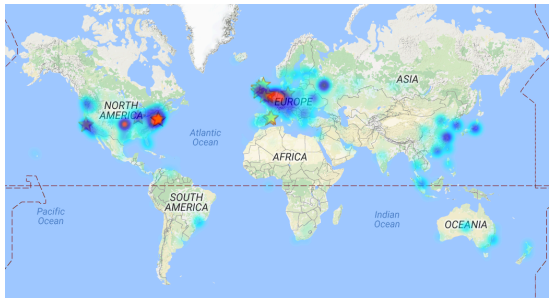
Purpose

An open, distributed computing platform, allowing custom applications to be created and run (*smart contracts* or *ĐApps*)



Topology

A public global network[1] of **~6,000** nodes



Features

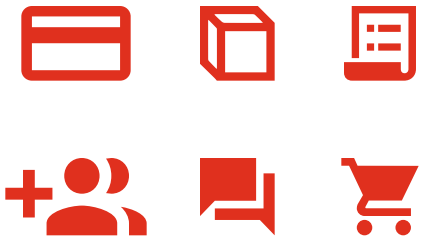
- ✅ No central point of failure
- ✅ Flexibility to run a wide variety of applications, with rich validation and execution logic (*turing-complete language*)
- ✅ Smart contracts can run as autonomous agents

[1] **Ethernodes** -http://ethernodes.org/network/1

Example - *Eris*

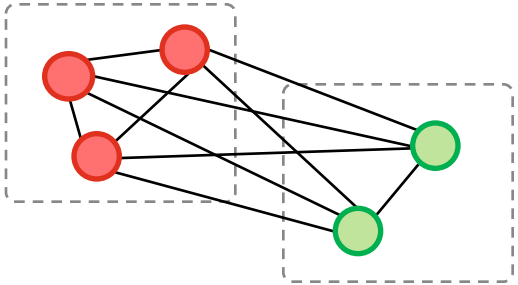
Purpose

A blockchain *fabric* to create private and semi-private (consortium) blockchains



Topology

Flexible - can be deployed in a variety of configurations



Features

- ✅ Increased privacy
- ✅ Ability to configure & govern how the blockchain operates (bring your own rules)
- ✅ Rich validation & execution logic
- ✅ Cheaper transactions
- ✅ Custom permissions model and swappable consensus


Why would I want one?

“

Blockchain’s transparency, security, and efficiency make it a particularly good choice for ***reshaping businesses*** that are bogged down by inefficiencies

GOLDMAN SACHS

Blockchains can **reduce costs and complexity**

Blockchains can **reduce errors**

Blockchain has **proven security**

Blockchains can **aid in auditability**

Blockchains have **proven resilience**

Blockchains create **shared trusted transactions**

Consider it when…

- ✅ Multiple parties want to ***share*** data
- ✅ Multiple parties need to ***update/add*** data
- ✅ Participants are ***non-trusting***
- ✅ Intermediaries add ***cost*** and ***complexity***
- ✅ Interactions are time sensitive

Be aware of the challenges

- Objectively evaluating the use-case is critical
- Blockchain is still a nascent field
- Key management is critical
- Immutable means forever!
- Participation ≠ voluntary full disclosure
- Confidentiality & privacy must be carefully considered
- Digital identities of actors need to be adequate, binding and non-repudiable in the real world
- This is not just about technology – business process redesign is likely

PwC Blockchain Delivery Group

Who are we?

- Product & Design Thinking Lead
- Deep domain expertise in technology and software engineering
- **150+ years** experience building and running platforms in the FinTech arena
- Extensive experience in designing and delivering innovative solutions
- Proven record in delivering **business critical enterprise** platforms
- Built the world's largest credit card processing platform
- Built a bitcoin payment processor and other Blockchain capabilities

How can we help?

Strategy

- Understand overall business impact
- Discover & prioritise use cases
- Understand the impact on the legacy environment
- Develop adoption strategy
- Develop execution roadmap

Design

- Use case development
- Architectural design & product selection
- Business process definition
- Support policies, governance & controls
- Go-live assurance
- Benchmarking & stress testing
- Business continuity planning

Execution

- Establish blockchain lab
- Agile development of PoCs
- Skills transfer and training
- Consortia establishment and management
- Product implementation and integration

2016, so far

Zyen / London Market Group

Policy Placement PoC

Technologies

- MultiChain
- Docker, Bootstrap
- GitHub, Trello, Slack, NodeJS

Description

Smart contract blockchain solution for the wholesale insurance markets, allowing syndicates, brokers to negotiate offers with regulatory oversight.

Bank of England

Cloud Blockchain prototype

Technologies

- Amazon AWS, Ethereum
- Docker, Docker Swarm / Hub
- Meteor, Bamboo, Trello

Description

Creation of a blockchain solution for banking on a cloud platform with the Bank of England.

Major Insurance Brokerage

Claims Management PoC

Technologies

- Eris Industries (Monax)
- Docker, Bootstrap, Backbone
- NodeJS, IPFS, GitHub, Trello

Description

Enabling the creation of claims and attachment of invoices to which can be approved or declined by participants on a blockchain.

Major Global Banking Institution

Smart Loans

Technologies

- On-Premise Technologies
- Eris Industries
- Docker

Description

Initial phase of digitisation of the bank's commercial loan process

Leading Global Software Company

Trade Finance PoC

Technologies

- Microsoft Azure
- Ethereum
- Docker

Description

Digitisation of one of the largest internal company treasury's, globally. Focus on SBLC Trade Finance process - spanning corporates, banks and parent company.

PwC Internal Projects

PwC Coin

Technologies

- Amazon AWS
- Ethereum
- Docker

Description

Proof of concept demonstrating asset (PwC Coin) transfer, interest payments, minting and regulator view of transactions.


And….relax

Thank you!

Questions?