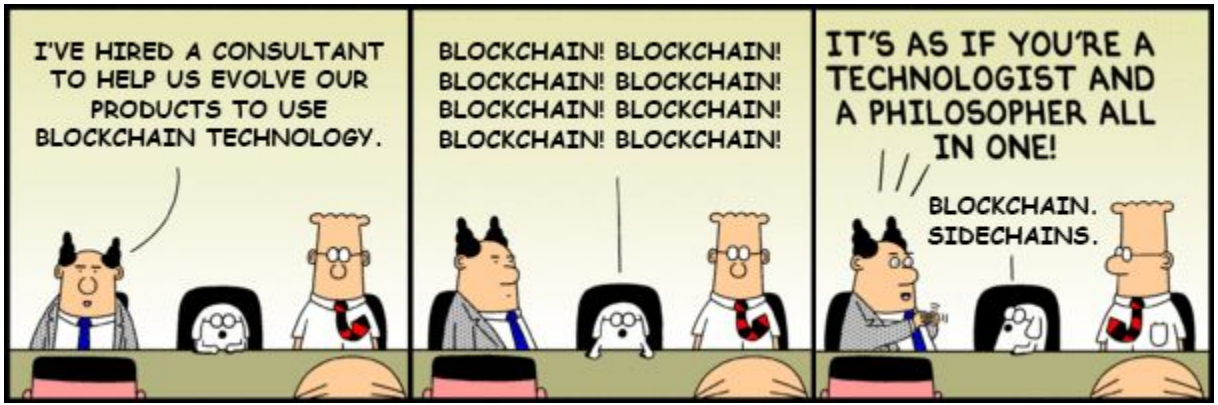
Leveraging Blockchain technologies For the Internet of Things

Sam BESSALAH - @samklr

Me :

- **Freelance Data Engineer, based in Paris and Amsterdam**
- **Passionate about all things distributed and machine learning.**
- **Tinkering with Blockchain, Bitcoin, Ethereum at night**
- **Definitely not a blockchain expert !!!**


I THINK WE SHOULD BUILD A BLOCKCHAIN
UH-OH
DOES HE UNDERSTAND WHAT HE SAID OR IS IT SOMETHING HE SAW IN A TRADE MAGAZINE AD?
WHAT COLOR DO YOU WANT THAT BLOCKCHAIN?
I THINK MAUVE HAS THE MOST RAM.
S Adams
E-mail: SCOTTADAMS@AOL.COM
© 1995 United Feature Syndicate, Inc.(NYC)


I'VE HIRED A CONSULTANT TO HELP US EVOLVE OUR PRODUCTS TO USE BLOCKCHAIN TECHNOLOGY.
BLOCKCHAIN! BLOCKCHAIN! BLOCKCHAIN! BLOCKCHAIN! BLOCKCHAIN! BLOCKCHAIN! BLOCKCHAIN! BLOCKCHAIN!
IT'S AS IF YOU'RE A TECHNOLOGIST AND A PHILOSOPHER ALL IN ONE!
BLOCKCHAIN. SIDECHAINS.


Bloomberg Markets
IT'S ALL ABOUT THE
BLOCKCHAIN
BLYTHE MASTERS IS BETTING THE DIGITAL LEDGER BEHIND BITCOIN WILL REVOLUTIONIZE THE WAY WE TRADE BONDS, LOANS, DERIVATIVES, AND MUCH ELSE.
THE INNOVATORS
BRAD KATSUYAMA'S NEXT CHAPTER
A BANK FOR PEOPLE WHO HATE BANKS
WHO WANTS TO START AN ETF?
A HUNDRED APPS BLOOM IN CHINA


INSIDE: A 12-PAGE SPECIAL REPORT ON COLOMBIA
The Economist
Our guide to America's best colleges
Myanmar's free-ish election
Those ever-creative accountants
America takes the fight to IS
Coywolves: the new superpredator
The trust machine
How the technology behind bitcoin could change the world

Bitcoin

> *A purely peer-to-peer version of electronic cash [which] would allow online payments to be sent directly from one party to another without going through a financial institution.*

Satoshi Nakamoto, in the Bitcoin Paper published in 2008?

Bitcoin

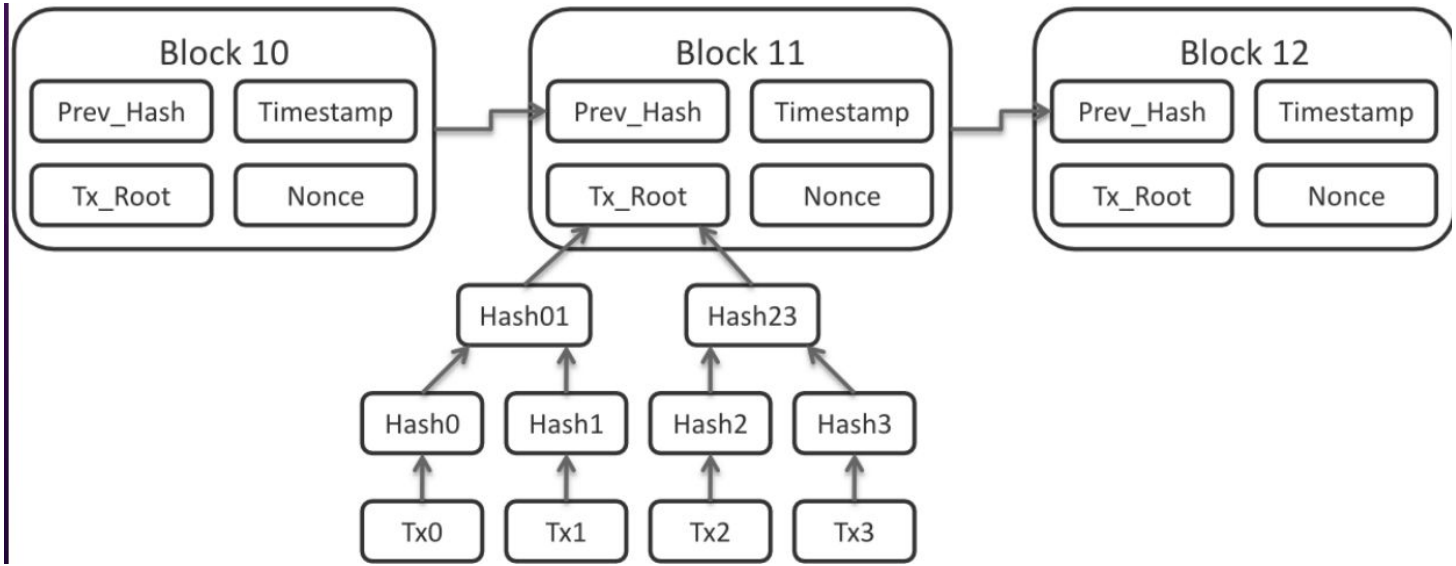
> *A purely peer-to-peer version of electronic cash [which] would allow online payments to be sent directly from one party to another without going through a financial institution.*

In practice, it is a combination of many things :

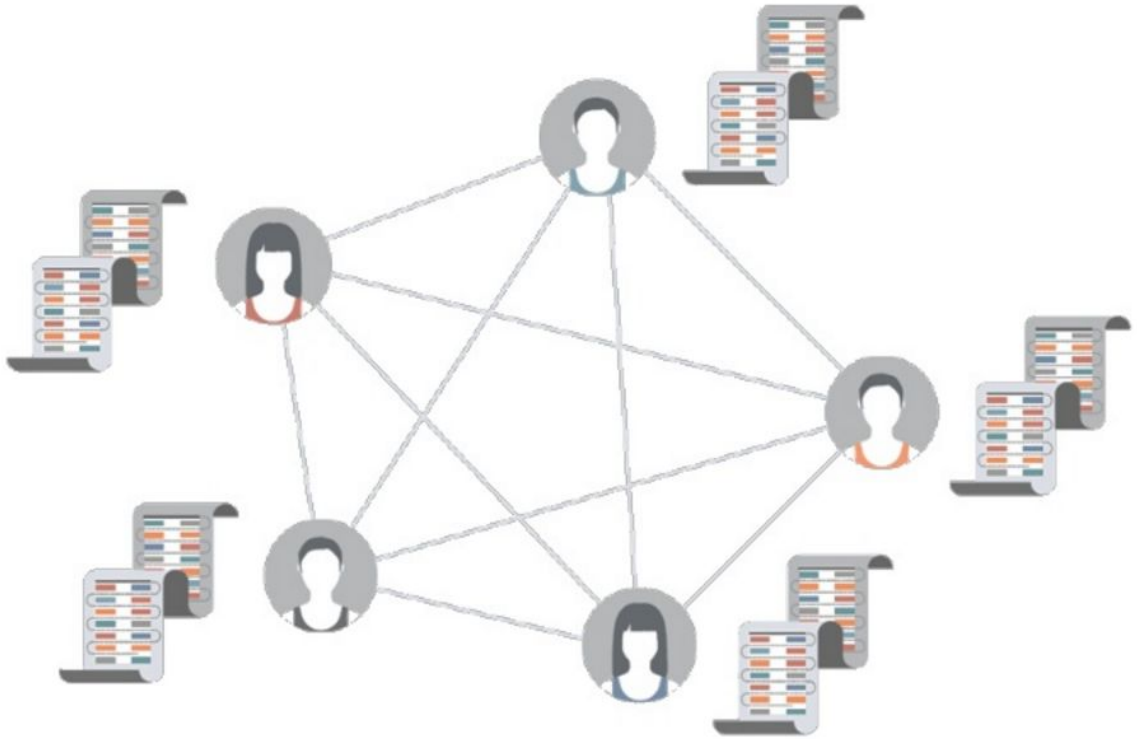
- **A P2P architecture like BitTorrent and Public Cryptography**
- **A “secure network” where transactions can be independently confirmed as unique and valid without a central authority : “The Blockchain”**

Blockchain

- The open source, decentralized ledger of all transactions, which is the backbone of the bitcoin network.
- Built of blocks (batches of transactions) in a sequential log or ledger of all transactions.

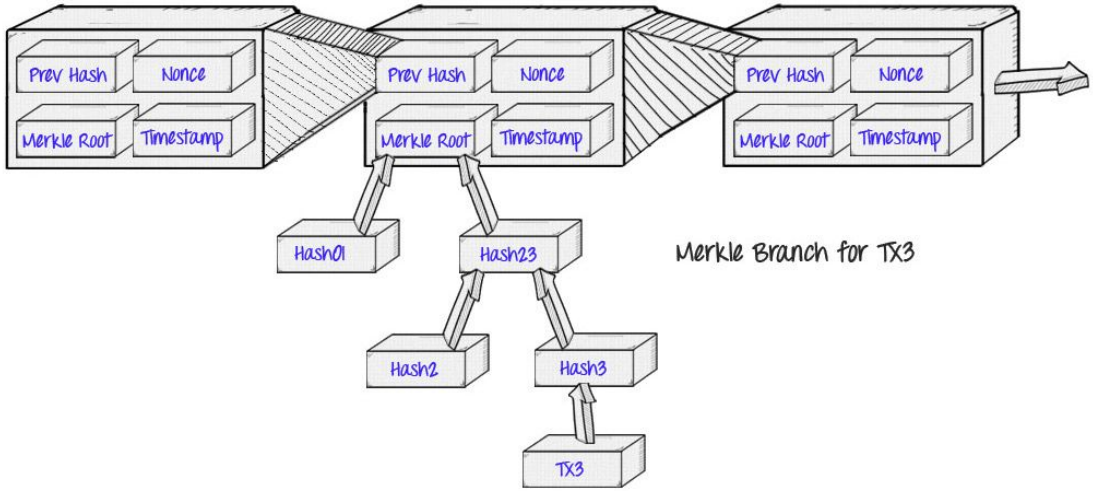


Blockchain



A copy of the blockchain is stored on each user's computer.

Blockchain



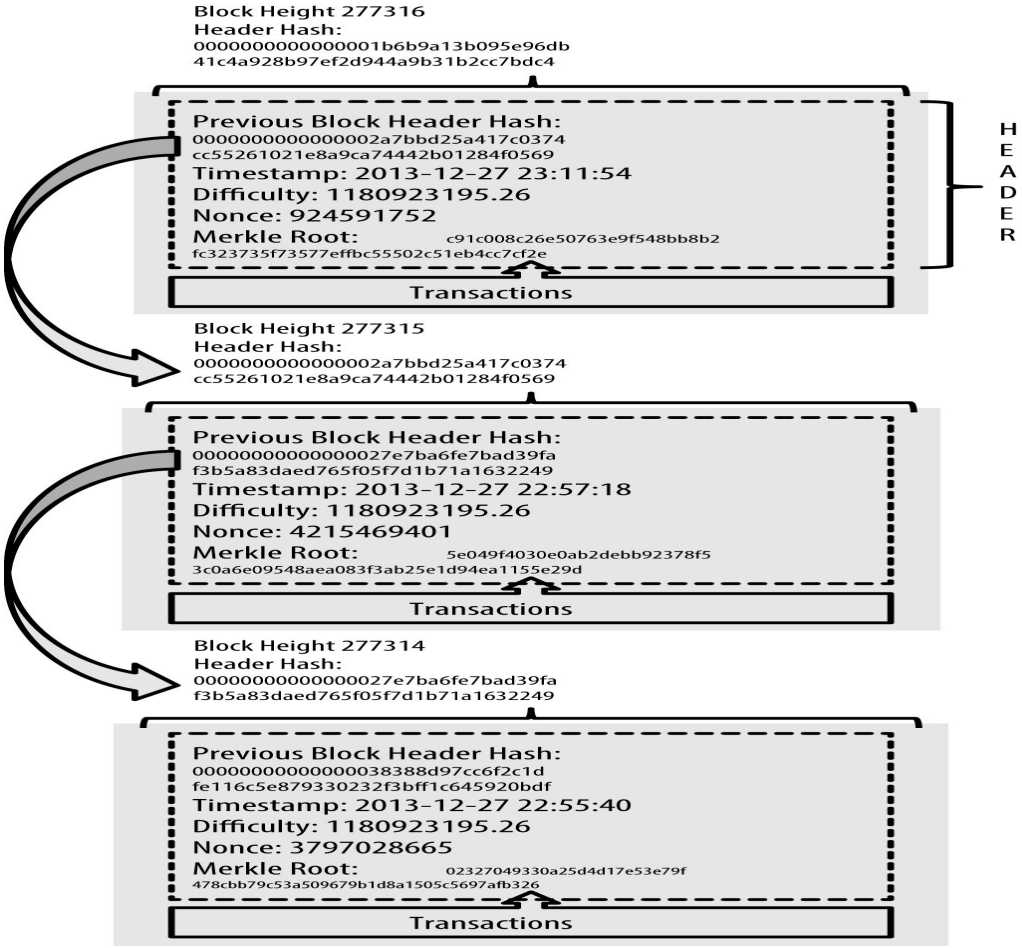
Adding a transaction

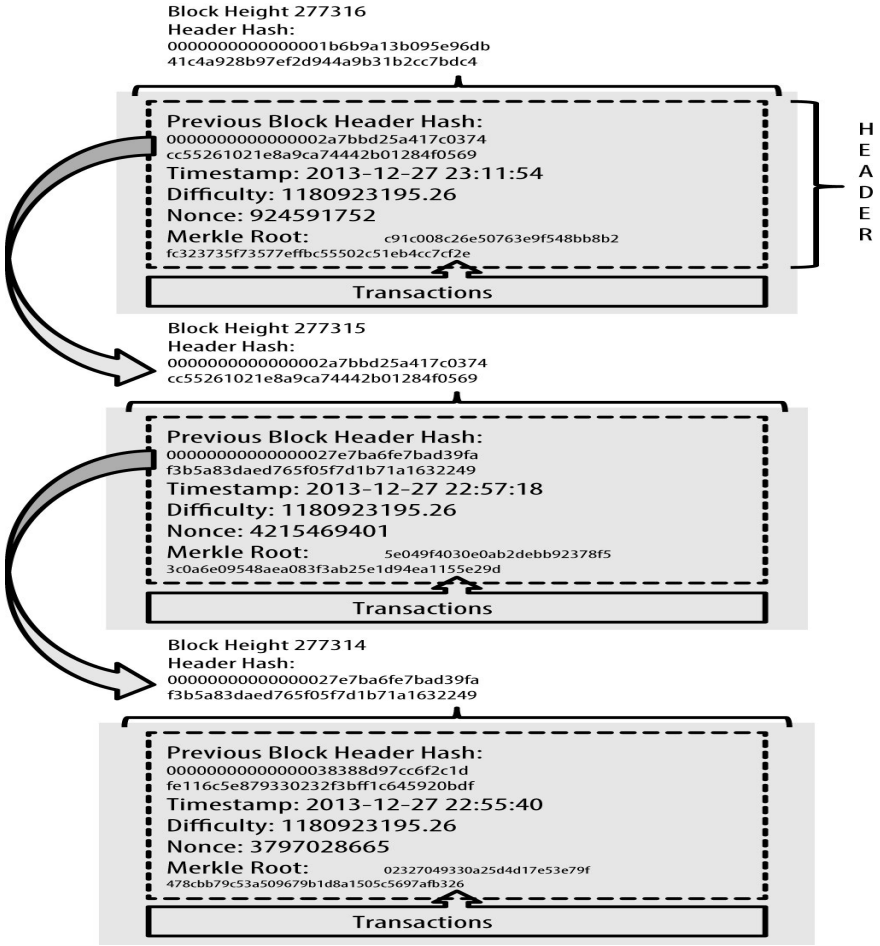
1 - New Transactions are broadcasted

2 - Every node creates creates a block, with seen transactions from his chain.

3- A node is “randomly” picked, and this node broadcasts its block

4- The other nodes receive and validate this block, and if it’s validated they add the block to their copy of the chain.


Block Height 277316
Header Hash:
000000000000000001b6b9a13b095e96db
41c4a928b97ef2d944a9b31b2cc7bdc4
Previous Block Header Hash:
000000000000000002a7bbd25a417c0374
cc55261021e8a9ca74442b01284f0569
Timestamp: 2013-12-27 23:11:54
Difficulty: 1180923195.26
Nonce: 924591752
Merkle Root: c91c008c26e50763e9f548bb8b2
fc323735f73577effbc55502c51eb4cc7cf2e
Transactions
HEADER
Block Height 277315
Header Hash:
000000000000000002a7bbd25a417c0374
cc55261021e8a9ca74442b01284f0569
Previous Block Header Hash:
000000000000000027e7ba6fe7bad39fa
f3b5a83daed765f05f7d1b71a1632249
Timestamp: 2013-12-27 22:57:18
Difficulty: 1180923195.26
Nonce: 4215469401
Merkle Root: 5e049f4030e0ab2debb92378f5
3c0a6e09548aea083f3ab25e1d94ea1155e29d
Transactions
Block Height 277314
Header Hash:
000000000000000027e7ba6fe7bad39fa
f3b5a83daed765f05f7d1b71a1632249
Previous Block Header Hash:
000000000000000038388d97cc6f2c1d
fe116c5e879330232f3bff1c645920bdf
Timestamp: 2013-12-27 22:55:40
Difficulty: 1180923195.26
Nonce: 3797028665
Merkle Root: 02327049330a25d4d17e53e79f
478cbb79c53a509679b1d8a1505c5697afb326
Transactions



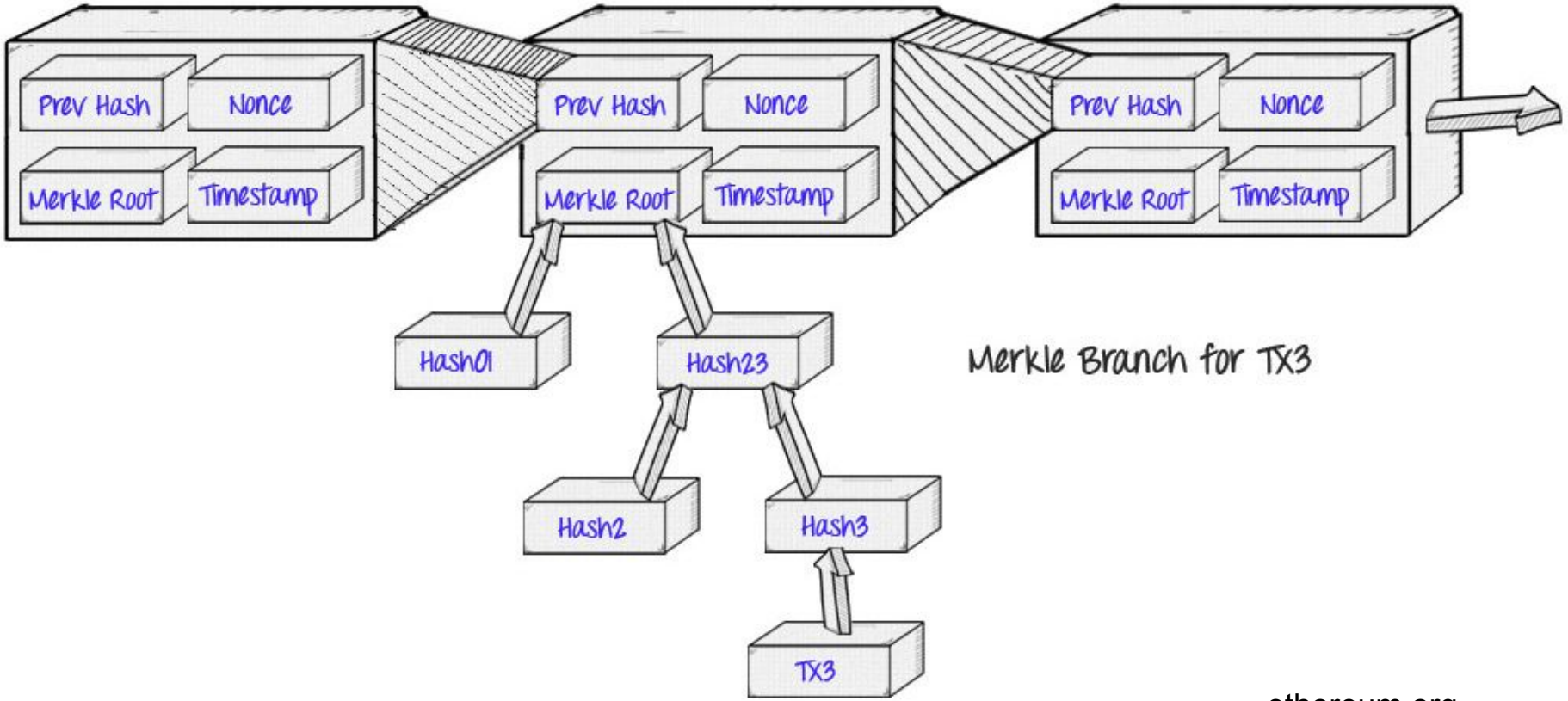
Merkle Tree

Allow for efficient summarizing, and verification of the integrity for large data sets. (Find them in distributed Databases like Cassandra), they contain cryptographic hashes.

In Bitcoin, they summarize all the transactions in a block, producing a digital fingerprint for a set of transactions, that is verifiable, and auditable.

It's built by hashing recursively hashing pairs of nodes until there's only one hash : the "merkle root".

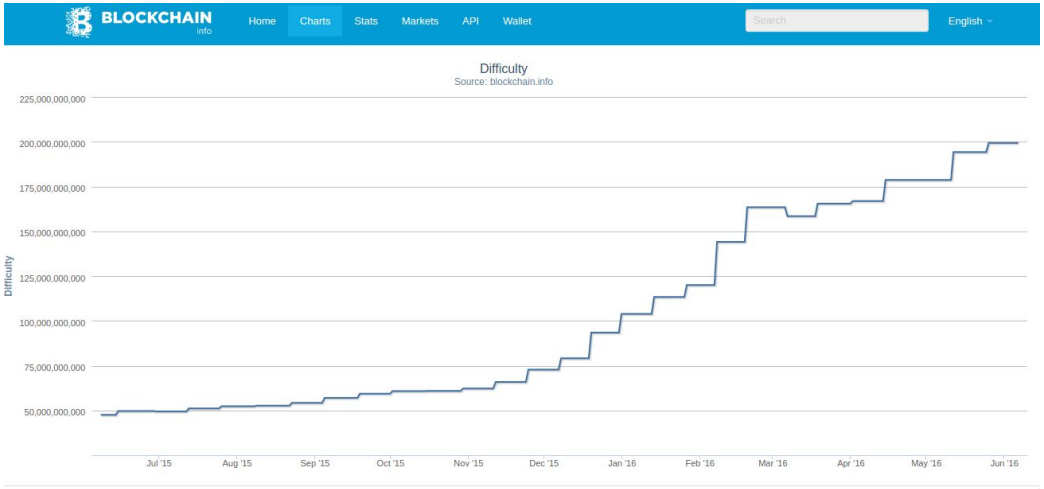
The cryptohash used here is a SHA256, applied twice.



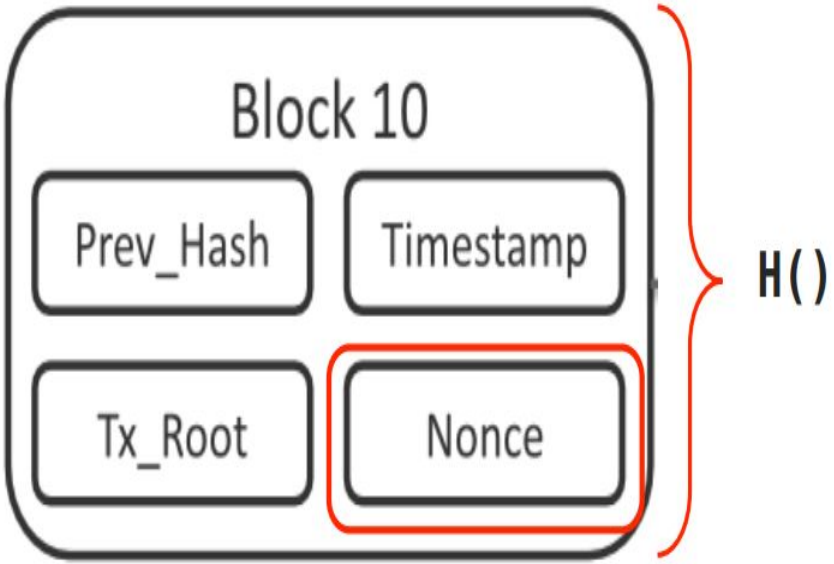
ethereum.org

Blockchain Decentralized Consensus

- **Bitcoin resolves consensus by “trying” to solve the famous “byzantine generals problem”, through the process of “Mining”**
- **Through the Mining process, Miners creates new blocks**
- **It amounts to solve an ever increasing challenge, but easy to verify by the others.**
- **Proof of work : A new block roughly added every *10 minutes***

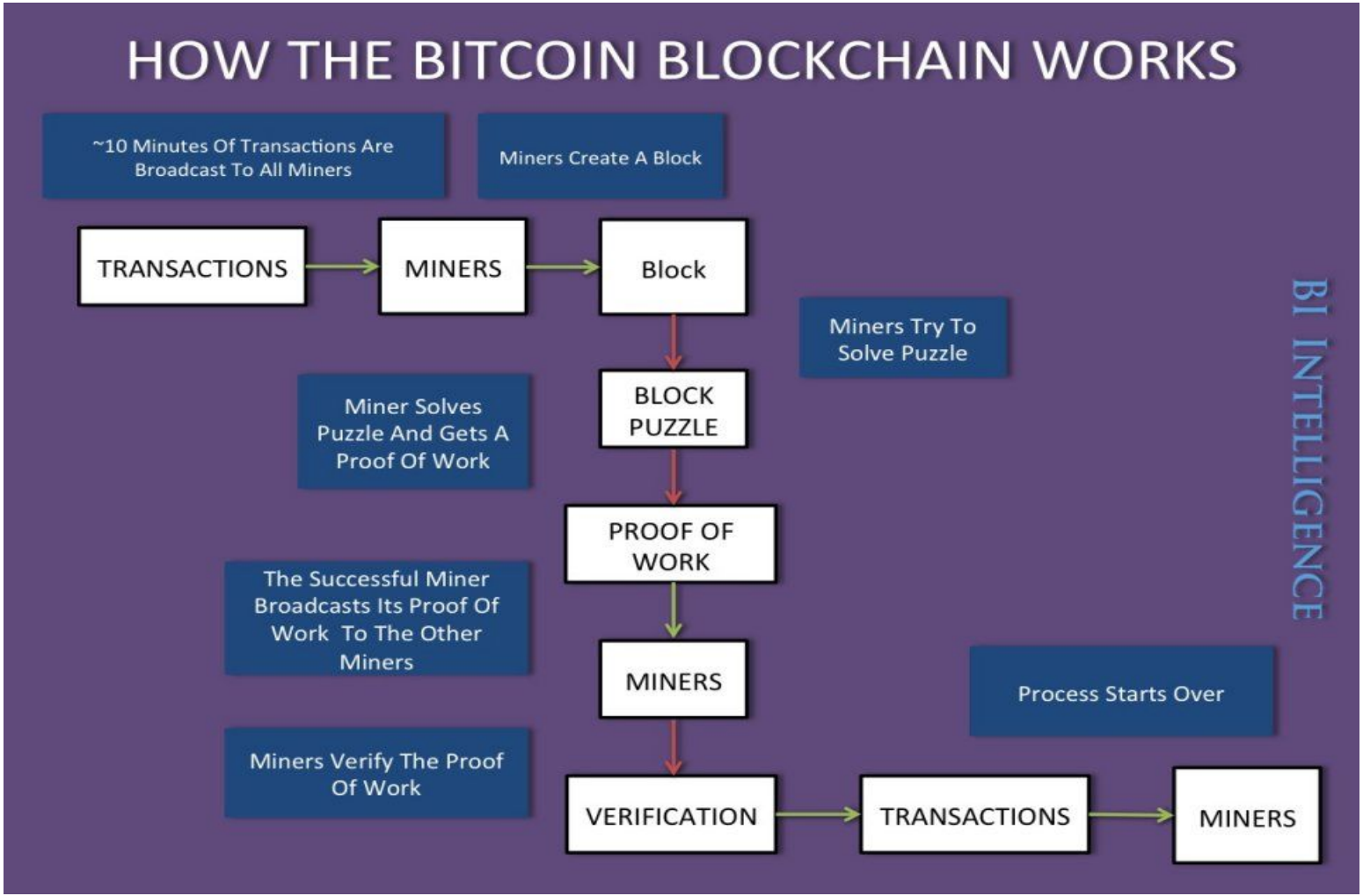


Blockchain : Proof of work



Find H where
H(_) < value

```
while block_hash > difficulty
   nonce = rand()
   block_hash = hash( block_header :: nonce )
```


HOW THE BITCOIN BLOCKCHAIN WORKS
~10 Minutes Of Transactions Are Broadcast To All Miners
Miners Create A Block
TRANSACTIONS
MINERS
Block
Miners Try To Solve Puzzle
BLOCK PUZZLE
Miner Solves Puzzle And Gets A Proof Of Work
PROOF OF WORK
The Successful Miner Broadcasts Its Proof Of Work To The Other Miners
MINERS
Process Starts Over
Miners Verify The Proof Of Work
VERIFICATION
TRANSACTIONS
MINERS
BI INTELLIGENCE


ethereum

Ethereum

- Bitcoin has sprung a lot of “altcoins” each with their Blockchain
- New type of blockchain that go beyond financial services
- Ethereum goes beyond most blockchains and aims at being versatile
- Contracts can be used to build currencies, voting systems, financial derivatives, decentralized applications, etc

Ethereum : Smart Contract

Ether is required to call a contract, this is done to ensure that infinite execution does not occur as the execution stops the moment the ether amount sent along with the call is consumed

When a contract is called, a small amount of ether is required to be sent along with the call. However, when a contract is called, an amount of ether can be directly transferred to that contract there by instituting a balance transfer like in the case of a payment

If a contract does not include any programmed instructions, it behaves as an account. It has an address and a balance. This account can receive/send ethers from/to other accounts/contracts.

Ethereum : Smart Contract Example

- Address : 0xc0D801061070FB92622d1d58Fe27872F07F0ea6
- Balance : 0
- Fields
  - Members (list of member)
  - nextMeetupDescription
  - nextMeetupDate
  - Votes
- Methods
  - Vote
  - AddMember
  - NewMeetupProposal
  - ExecuteProposal
- Events
  - Vote
  - AddMember
  - NewMeetupProposal
  - ExecuteProposal

Ethereum : Smart Contracts

LLL, Serpent, and **Solidity** all can be used to write contracts

```
contract owned {
    address public owner;

    function owned() {
        owner = msg.sender;
    }

    modifier onlyOwner {
        if (msg.sender != owner)
            throw;
        _
    }

    function transferOwnership(address newOwner)onlyOwner {
        owner = newOwner;
    }
}

contract EthereumMeetup is owned {
```

Modifier

Inheritance

```
struct MeetupProposal {
    string nextMeetupDescription;
    uint nextMeetupDate;
    uint votingDeadline;
    bool executed;
    bool proposalPassed;
    uint numberOfVotes;
    int currentResult;
    Vote[]votes;
    mapping(address => bool) voted;
}
```

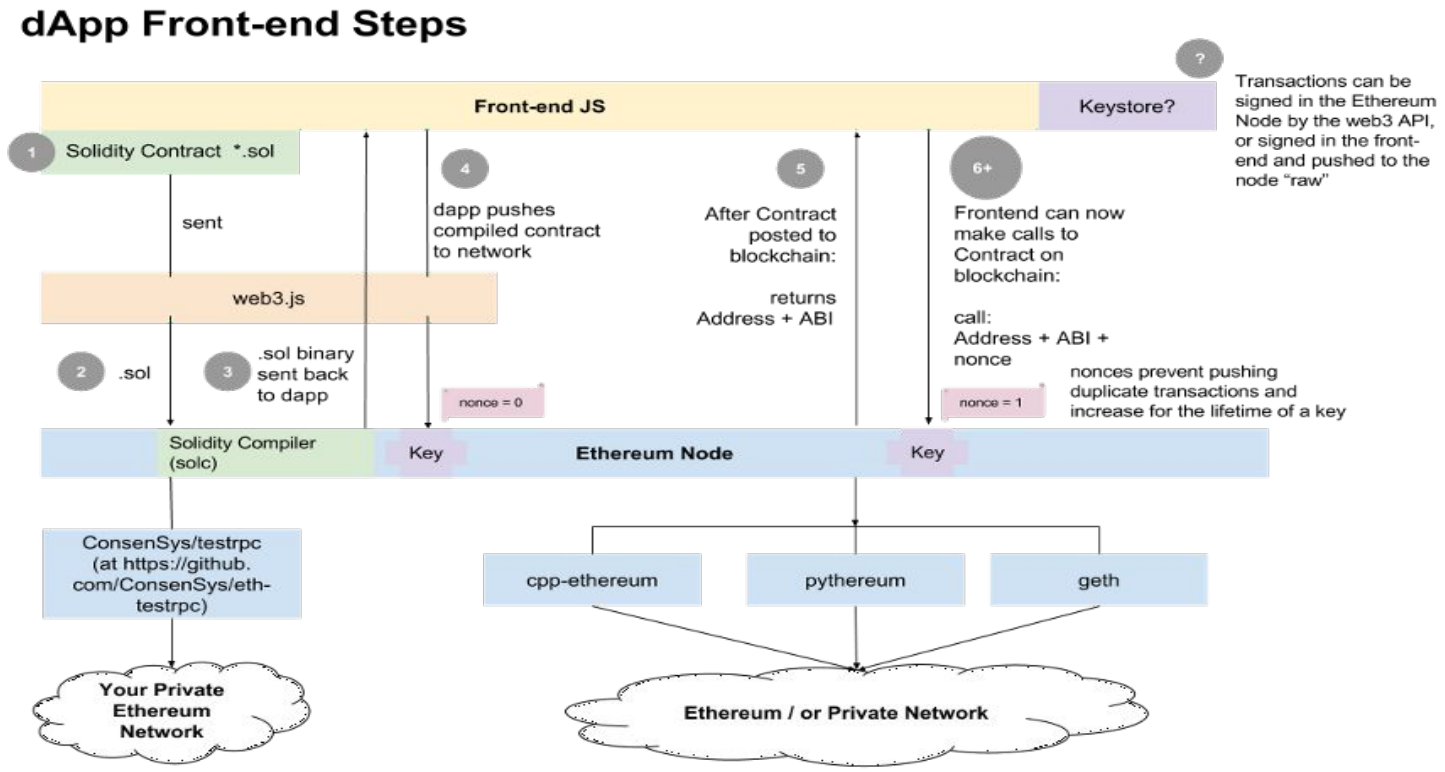
Hashtable

```
struct TransferFundProposal {
    address recipient;
    uint amount;
    string description;
    uint votingDeadline;
    bool executed;
    bool proposalPassed;
    uint numberOfVotes;
    int currentResult;
    bytes32 proposalHash;
    Vote[]votes;
    mapping(address => bool)voted;
}

struct Member {
    address member;
    bool canVote;
    string name;
    uint memberSince;
}

struct Vote {
    bool inSupport;
    address voter;
    string justification;
}
```

Ethereum : Decentralized Apps (Dapps)

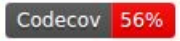


A **Contract Creation Transaction** is shown in steps 1-5 at above.

An **Ether Transfer** or **Function Call Transaction** is assumed in step 6.

Ethereum Go

Official golang implementation of the Ethereum protocol

	Linux	OSX	ARM	Windows	Tests
develop					 
master					 

Blockchain

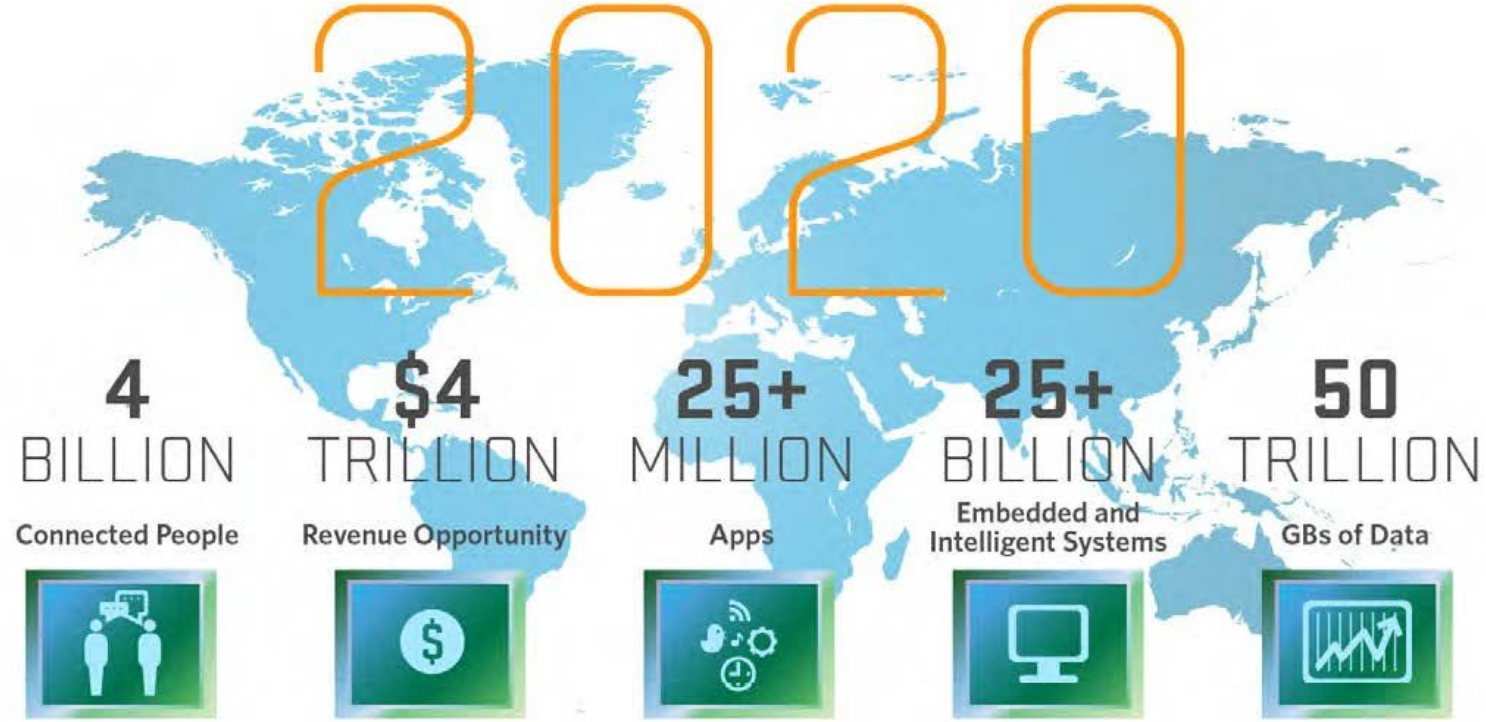
=

Secure

Transparent

Distributed Consensus

Connected Devices and IoT



Source: Mario Morales, IDC


canary


nest
IN 25 MIN
72

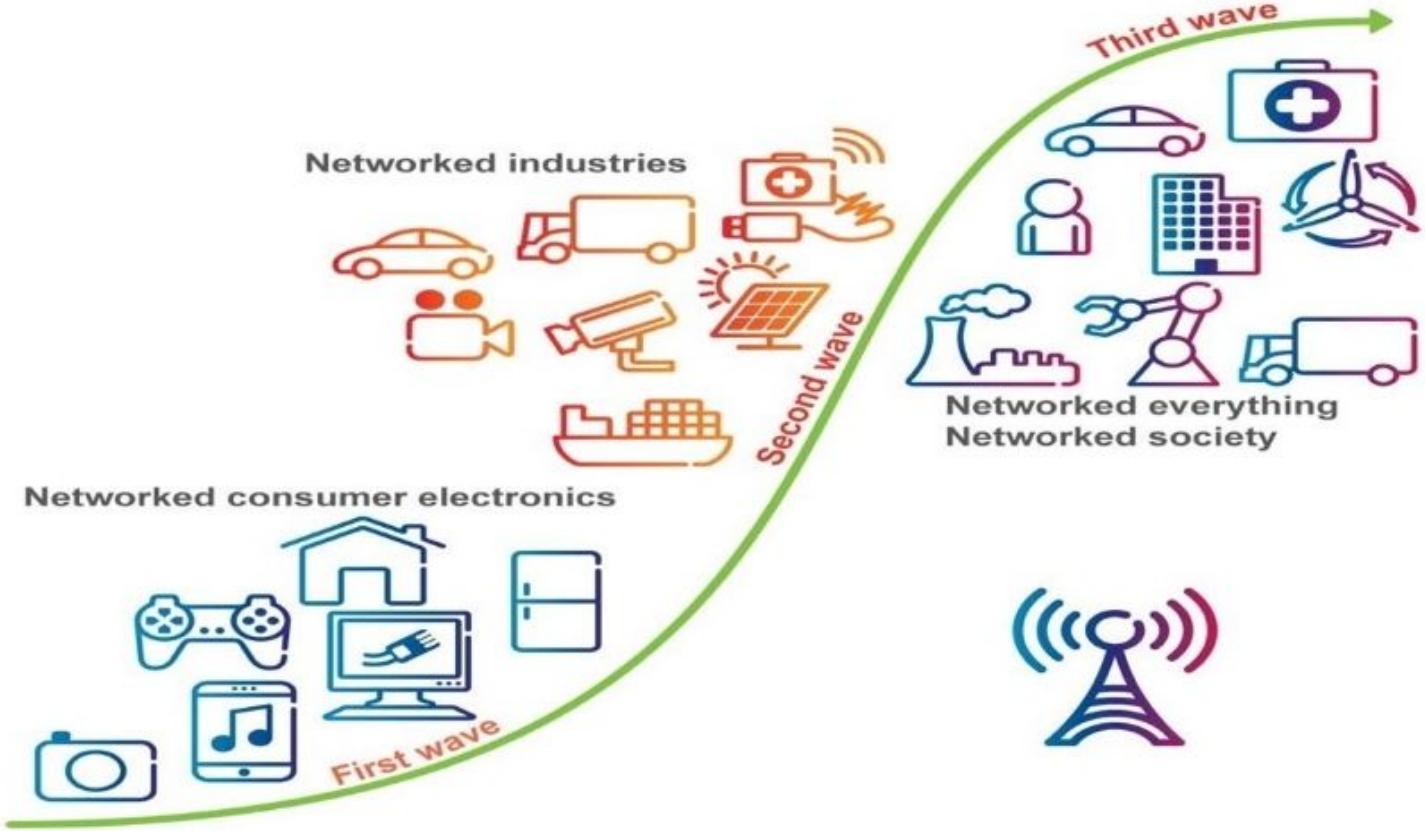

Parrot
Flower Power
Water
Your plant needs watering!

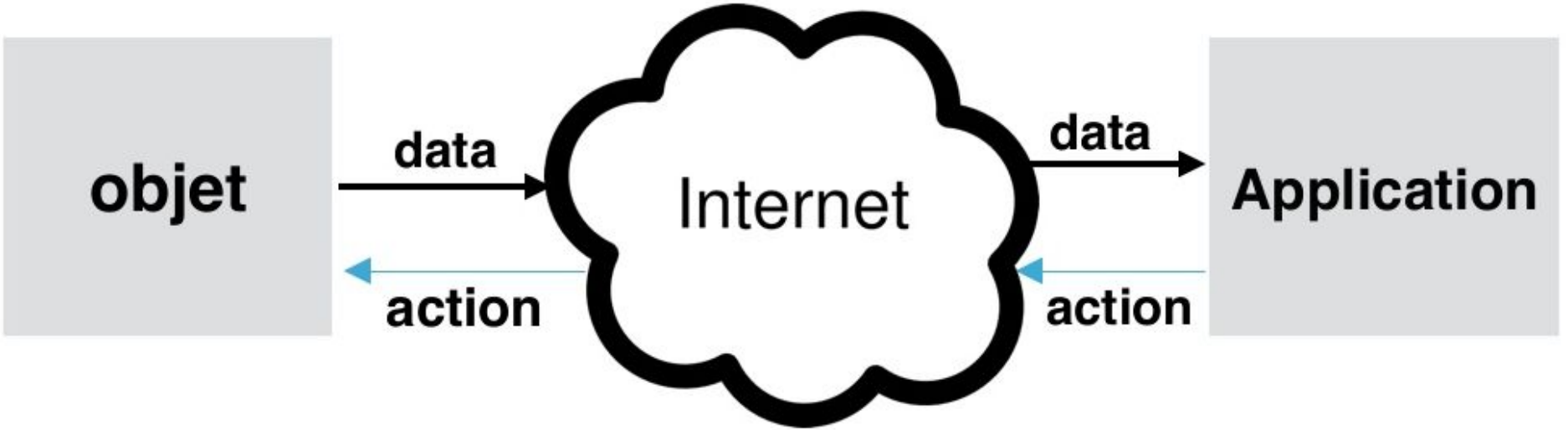
Nano Computer

RaspberryPi

Micro Controller

Arduino


Third wave
Networked industries
Second wave
Networked everything
Networked society
Networked consumer electronics
First wave

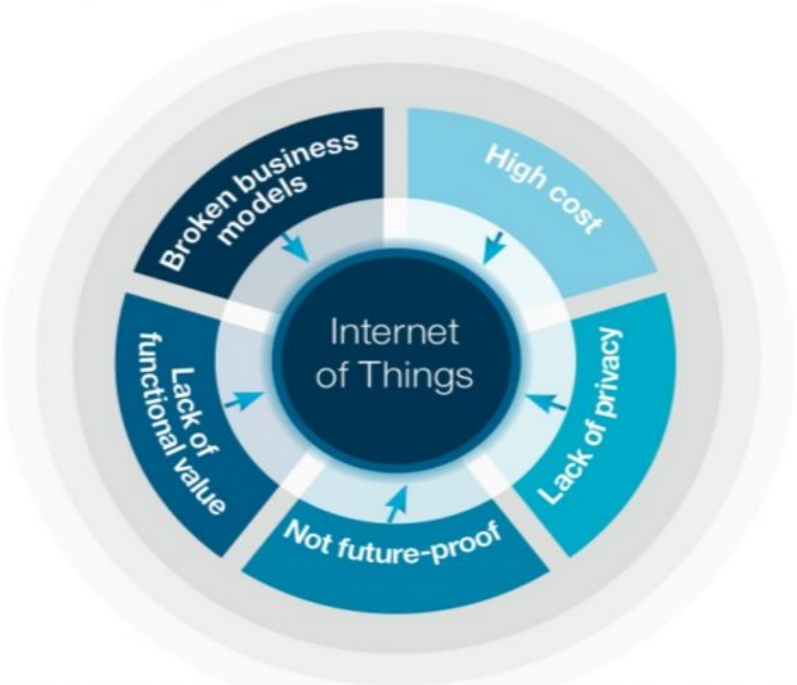

objet
data
action
Internet
data
action
Application

Security gaps in IoT with serious physical effects

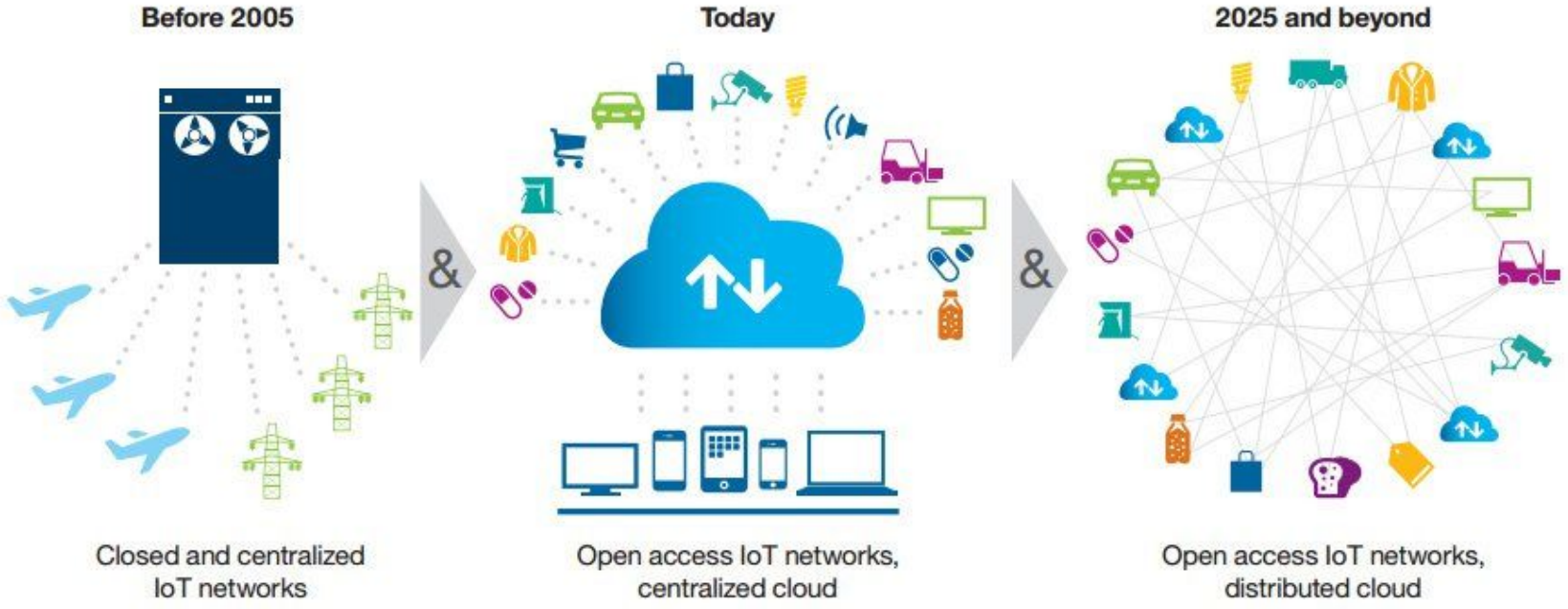
Refrigerator among devices hacked in Internet of things cyber attack

AFTER JEEP HACK, CHRYSLER RECALLS 1.4M VEHICLES FOR BUG FIX

Why today's Internet of billions of Things won't scale to the Internet of hundreds of billions of Things



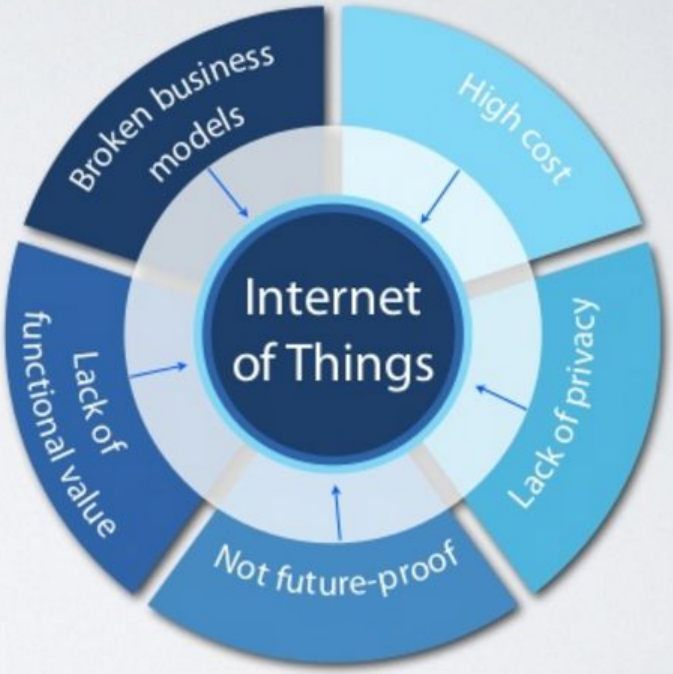
Source: Device democracy. Saving the future of the Internet of Things. IBM Institute for Business Value. http://www-935.ibm.com/services/us/gbs/thoughtleadership/internetofthings/ fig.2, p4


Before 2005
Today
2025 and beyond
&
&
Closed and centralized IoT networks
Open access IoT networks, centralized cloud
Open access IoT networks, distributed cloud

IBM and Samsung's ADEPT Project

ADEPT: IOT CHALLENGE FOCUS

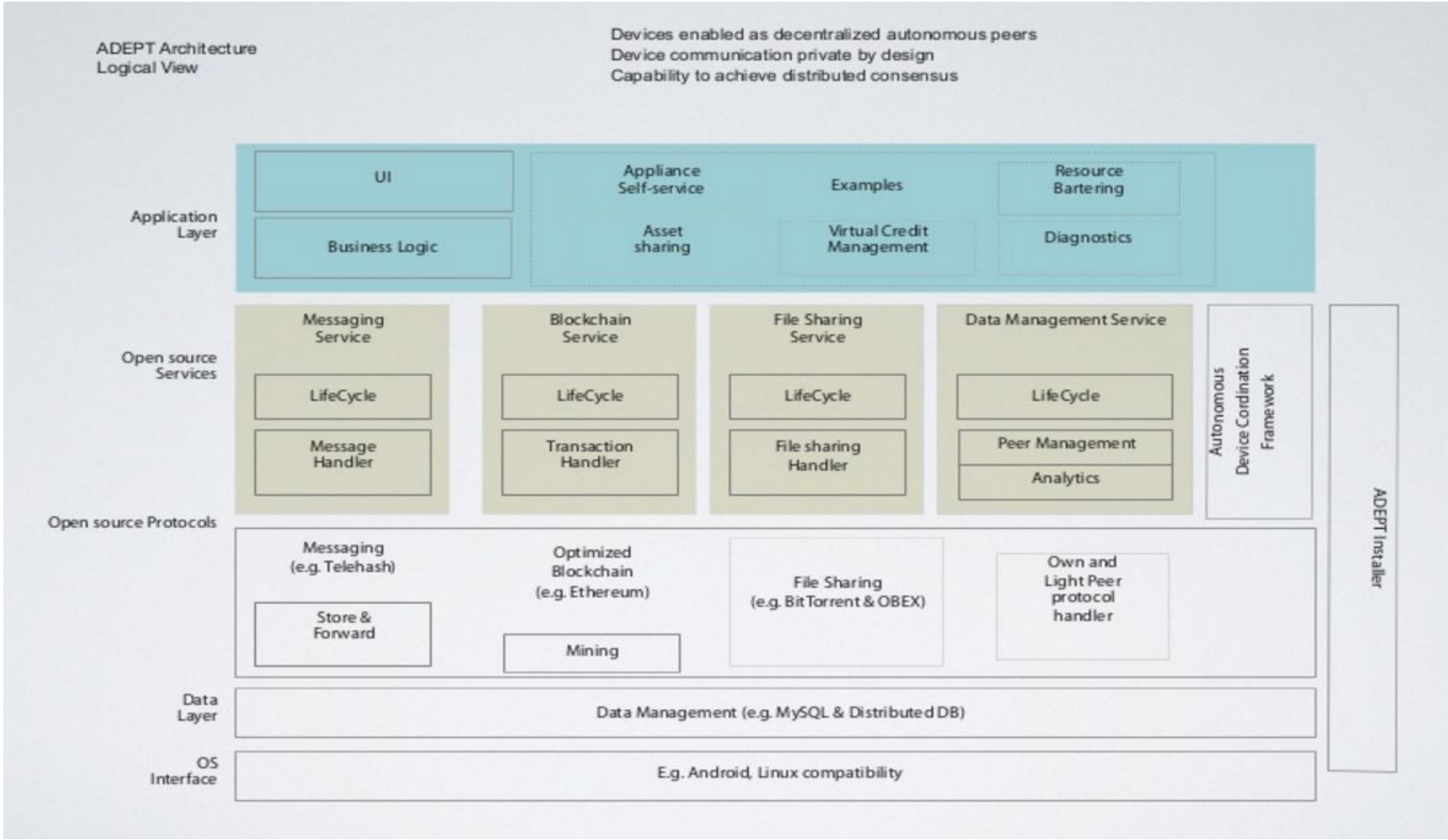
- √ cost
- ▫ trust
- √ monetisation
- √ interoperability
- √ discoverability
- √ authentication
- √ long term service expectation
- ▫ scale

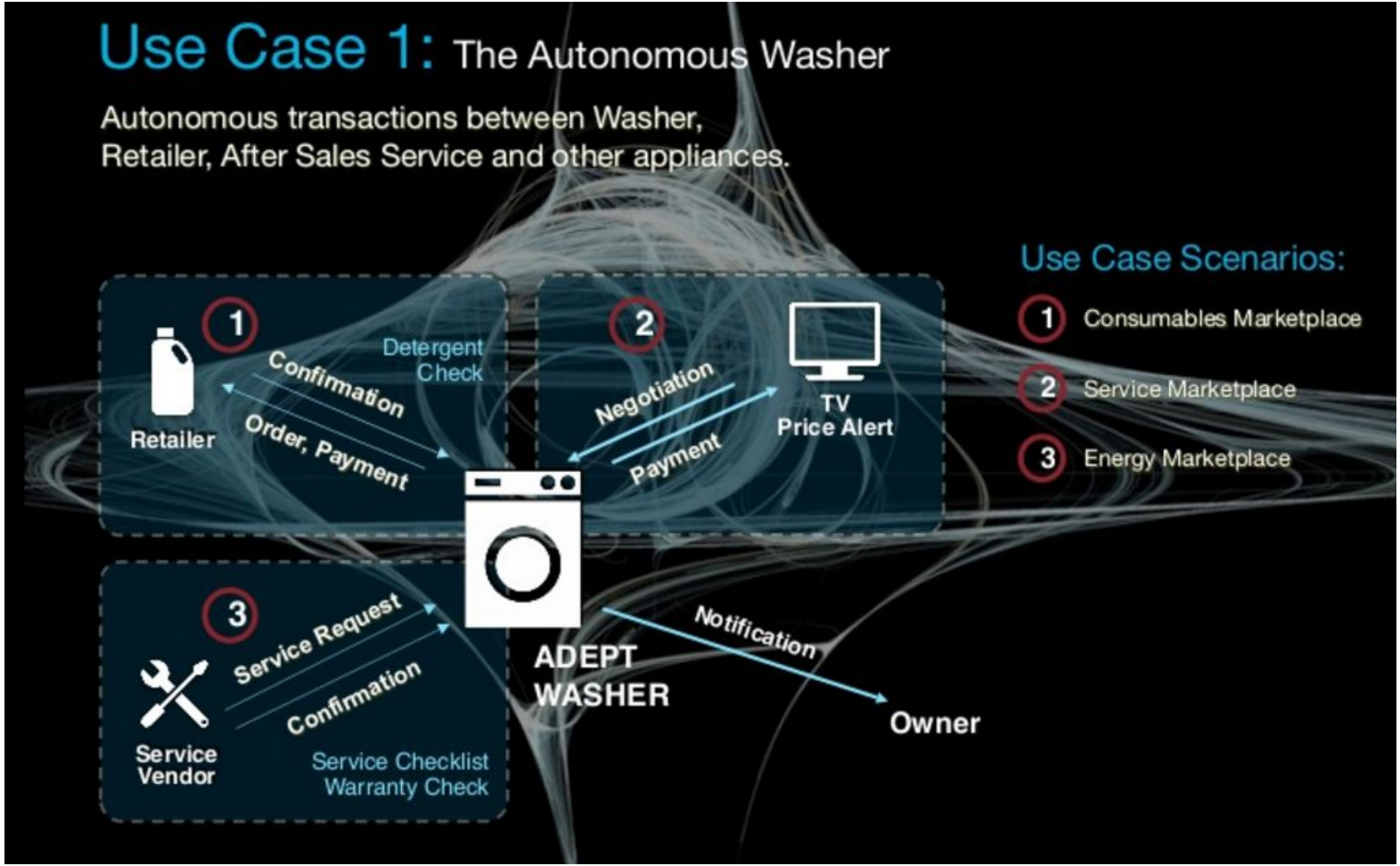


ADEPT VISION

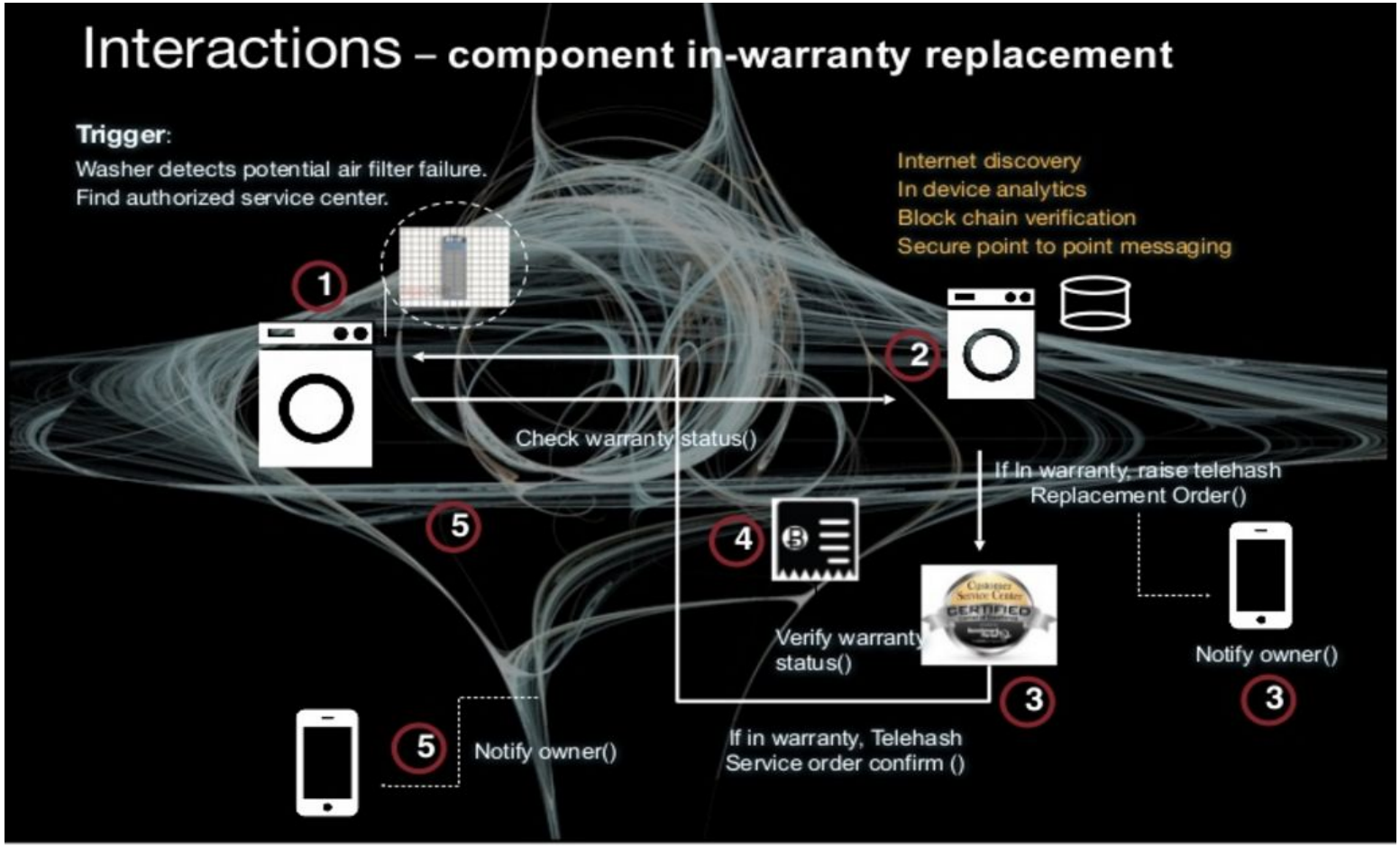
- ▫ highly efficient digital marketplaces
- √ real-time resource competition
- √ inter-device agreements
- √ direct payments between devices
- √ service and resource barter between devices
- ▫ inter-device reputation

√ = demo'ed

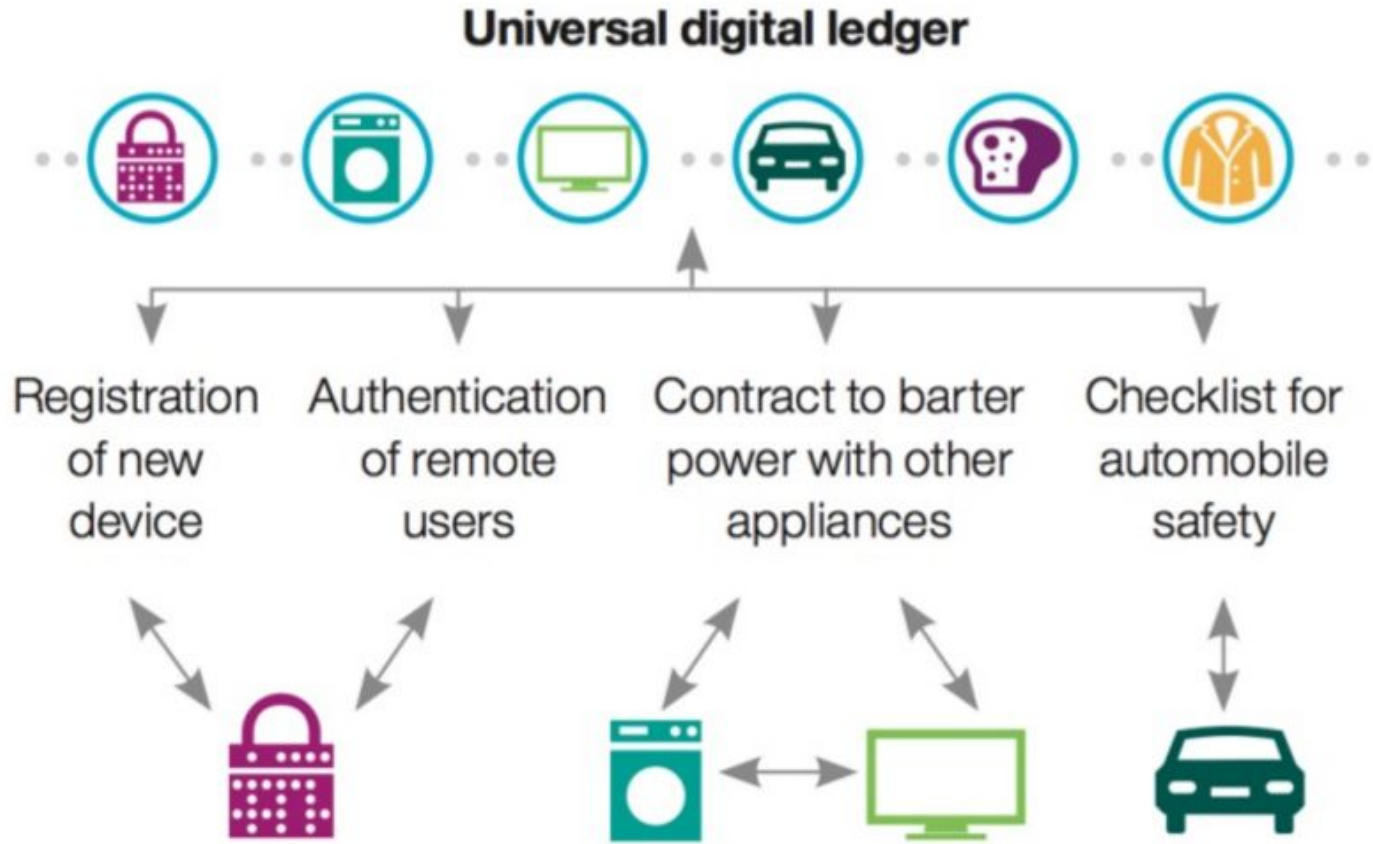

ADEPT Architecture
Logical View
Devices enabled as decentralized autonomous peers
Device communication private by design
Capability to achieve distributed consensus
Application Layer
UI
Business Logic
Appliance Self-service
Asset sharing
Examples
Virtual Credit Management
Resource Bartering
Diagnostics
Open source Services
Messaging Service
LifeCycle
Message Handler
Blockchain Service
LifeCycle
Transaction Handler
File Sharing Service
LifeCycle
File sharing Handler
Data Management Service
LifeCycle
Peer Management
Analytics
Autonomous Device Cordination Framework
ADEPT Installer
Open source Protocols
Messaging (e.g. Telehash)
Store & Forward
Optimized Blockchain (e.g. Ethereum)
Mining
File Sharing (e.g. BitTorrent & OBEX)
Own and Light Peer protocol handler
Data Layer
Data Management (e.g. MySQL & Distributed DB)
OS Interface
E.g. Android, Linux compatibility


Use Case 1: The Autonomous Washer
Autonomous transactions between Washer,
Retailer, After Sales Service and other appliances.
1
Detergent
Check
Confirmation
Order, Payment
Retailer
2
Negotiation
Payment
TV
Price Alert
3
Service Request
Confirmation
Service
Vendor
Service Checklist
Warranty Check
ADEPT
WASHER
Notification
Owner
Use Case Scenarios:
1 Consumables Marketplace
2 Service Marketplace
3 Energy Marketplace

Interactions – component in-warranty replacement

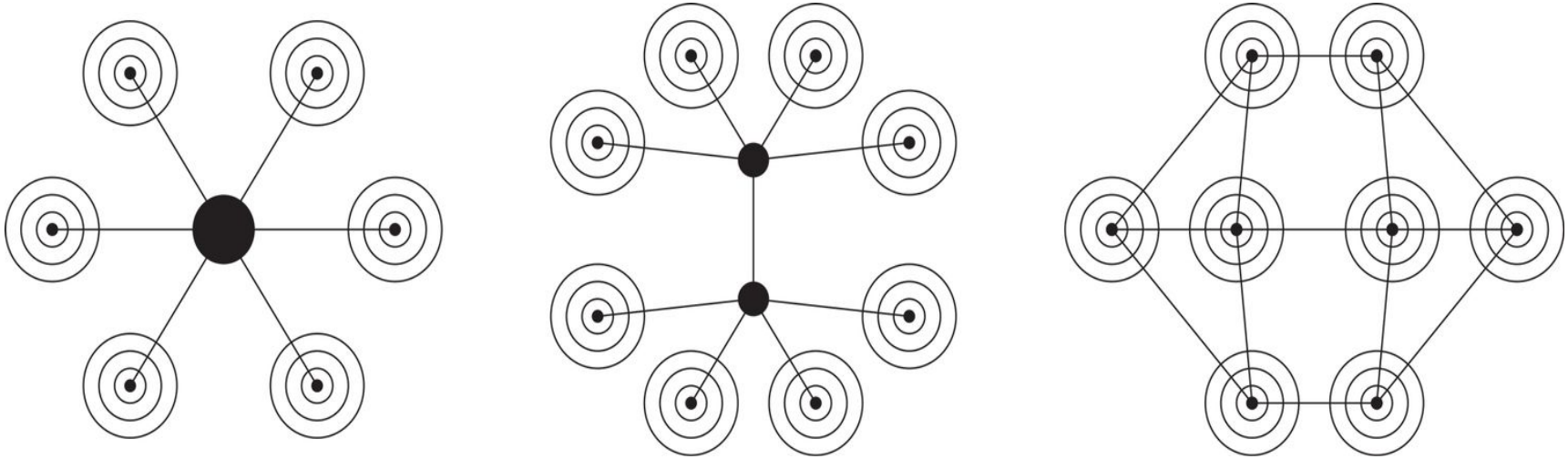


IBM

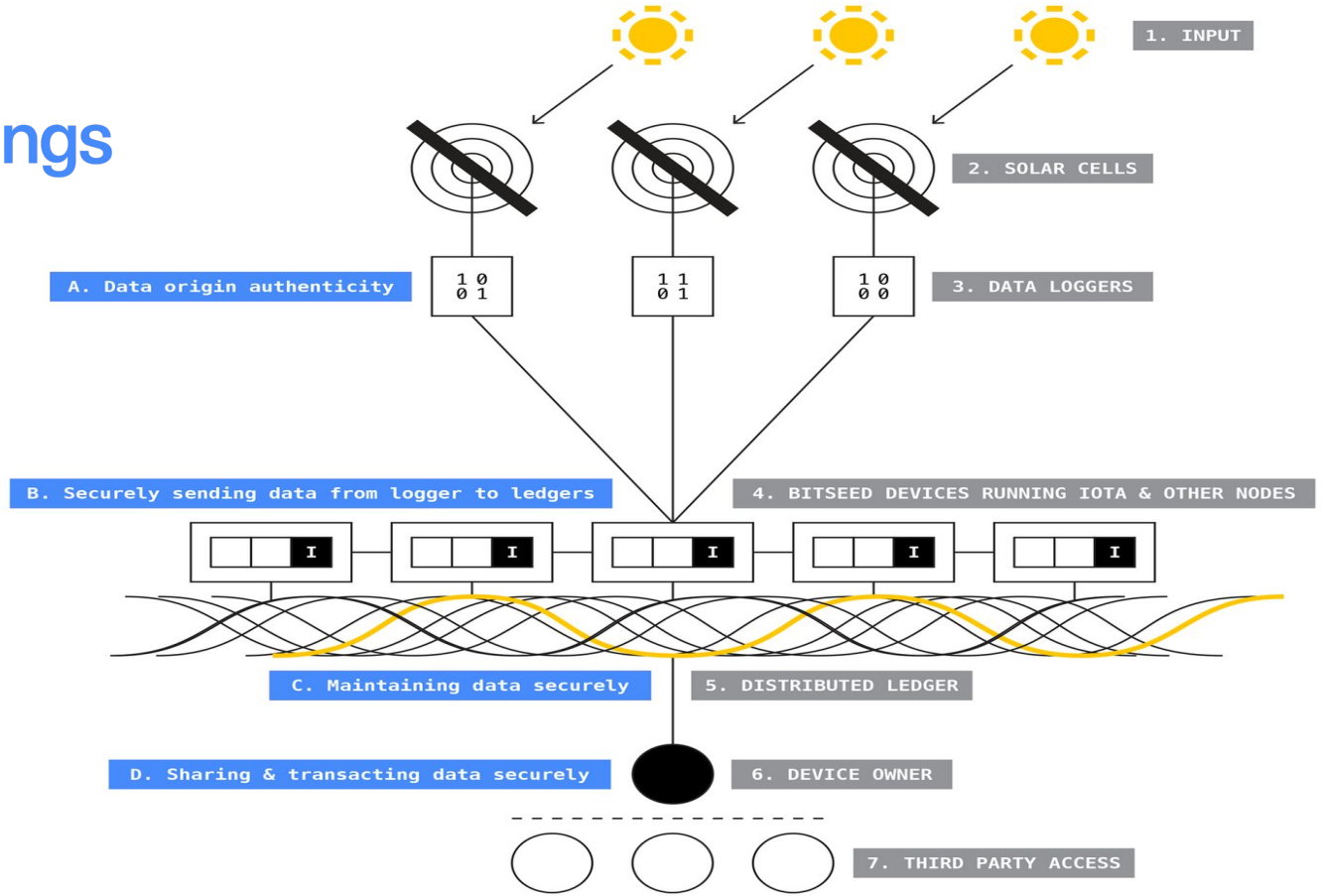


Source: Device democracy. Saving the future of the Internet of Things. IBM Institute for Business Value. http://www-935.ibm.com/services/us/gbs/thoughtleadership/internetofthings/ fig.5, p11

IoT Communications Architectures



chain
of
things



chain
of
things

The main security focus is on:

- Data origin authenticity
- Securely sending data from logger to ledgers
- Maintaining that data securely
- Sharing/transacting that data thereafter securely

Additional outstanding questions:

- How does this compare to legacy solutions?
- How will that data be controlled by its owner?
- How will the node know that it is receiving data from an approved datalogger?
- What is the most likely or secure way for the data to be relayed from the datalogger to the node?
- How many data loggers report back to a single node?

Getting Started, DIY IoT Blockchain

Getting Started, DIY IoT Blockchain

EthEmbedded

Raspi-Eth-Install

Welcome to Eth(Embedded)'s Raspbian based Ethereum Installer.

Click here for Raspbian Ethereum Installer on GitHub: https://github.com/EthEmbedded/Raspi-Eth-Install

- – Raspberry Pi agnostic – Wether you have a Pi 1B, 1B+, or Pi 2 it is the exact same process
- – Utilizes Raspbian(Debian based) OS for those of you who prefer that flavour:)
- – Unattended install of the OS
- – Consumes entire SD by default so no need to resize after install.
- – Utilize default settings OR customize by editing installer-config.txt file prior to install (root password, IP settings, hostname, etc.)
- – Utilizes dphys-swapfile so end user does not have to set up swap.
- – Choose your flavor of Ethereum by which script you run – eth/cpp-ethereum OR geth/go-ethereum

Slock.it

Code Example


"That's all Folks!"