Blockchain Explained
An Introduction to Blockchain for Business

Ángel Hernandez Bravo
Office of CTO Europe for Blockchain and National Security
Todo comenzó con ...

---

Bitcoin P2P e-cash paper

Satoshi Nakamoto | Sat, 01 Nov 2008 16:16:33 -0700

I've been working on a new electronic cash system that’s fully peer-to-peer, with no trusted third party.

The paper is available at:
http://www.bitcoin.org/bitcoin.pdf

The main properties:
- Double-spending is prevented with a peer-to-peer network.
- No mint or other trusted parties.
- Participants can be anonymous.
- New coins are made from Hashcash style proof-of-work.
- The proof-of-work for new coin generation also powers the network to prevent double-spending.

Bitcoin: A Peer-to-Peer Electronic Cash System
Todo comenzó con ...

Transactions

Transaction Chain

Block Chain

Distributed Shared Ledger
Ledgers are key

**Ledger** is THE system of record for a business. Business will have multiple ledgers for multiple business networks in which they participate.

– **Transaction** – an asset transfer onto or off the ledger
  - John gives a car to Anthony (simple)

– **Contract** – conditions for transaction to occur
  - If Anthony pays John money, then car passes from John to Anthony (simple)
  - If car won't start, funds do not pass to John (as decided by third party arbitrator) (more complex)
Business networks, wealth and markets

- **Business Networks** benefit from connectivity
  - Participants are customers, suppliers, banks, partners
  - Cross geography & regulatory boundary

- **Wealth** is generated by the flow of goods & services across business network in transactions and contracts

- **Markets** are central to this process:
  - Public (fruit market, car auction), or
  - Private (supply chain financing, bonds)
Problem...

... ineficiente, caro, vulnerable
A shared, replicated, permissioned ledger …

… with consensus, provenance, immutability and finality
Blockchain underpins Bitcoin

**bitcoin** is:

- An unregulated shadow-currency
- The first blockchain application
- Resource intensive

Blockchain for business differs in key areas:

- Identity over anonymity
- Selective endorsement over proof of work
- Assets over cryptocurrency
Shared ledger

- Shared between participants
- Participants have own copy through replication
- Permissioned, so participants see only appropriate transactions
- THE shared system of record

Records all transactions across business network
Frictions framework
Information, interaction and innovation frictions challenge business efficiency

1 Information
- Imperfect information
- Inaccessible information
- Information risks

Frictions

2 Interaction
- Transaction costs
- Degrees of separation
- Inaccessible marketplaces

3 Innovation
- Restrictive regulations
- Institutional inertia
- Invisible threats

Blockchain attributes framework
Five attributes of blockchains are key to reducing business frictions

- Distributed and sustainable
- Secure and indelible
- Transparent and auditable
- Consensus-based and transactional
- Flexible and orchestrated
Atributos Esenciales

Registro Distribuido (“append only”)
Mecanismo de Consenso
Procedencia
Inmutabilidad
Finalidad
¿Un modelo de blockchain específico para soluciones empresariales?

Permissionless vs Permissioned
Una plataforma tipo Bitcoin Blockchain ¿suficiente?

¿Rendimiento y escalabilidad?
¿Puedo adaptarlo a mis necesidades?
¿Y si queremos tener varias redes blockchain?
¿Por qué deben verlo todos?

Consumo energético
Interacciones Humanas = Contratos Complejos
Estandarización Multi-Industrial

......... Aparece el regulador
Un Blockchain empresarial (Reqs)

Basado en estándares
Registro e identidad
Privacidad y Confidencialidad (Datos y Contratos)
Auditable
Integrable con los sistemas tradicionales
Soporte avanzado a Contratos
Un Blockchain empresarial: (Reqs 2)

Alto rendimiento y escalabilidad
Algoritmos de Consenso
Energéticamente eficiente
Interoperable (multi-blochchains, multi-industria)
Plataforma flexible y extensible
Cloudificable
Un Blockchain empresarial

**HYPERLEDGER PROJECT**

- **MEMBERSHIP**
  - Identity, Privacy and Auditability of blockchain participants.

- **BLOCKCHAIN | TRANSACTIONS**
  - Distributed transaction ledger whereby the ledger is updated by consensus.

- **CHAIN-CODE**
  - "Smart Contracts", provide ability to run business logic against the blockchain.

- **APIs, SDKs, CLI**
  - Gives developers the ability to programmatically control the blockchain network.

**Blockchain APIs, SDKs, CLI**

**Membership Services**
- Registration
- Id. Management
- Auditability

**Blockchain Services**
- Consensus Manager
- Distributed Ledger
- P2P Protocol
- Ledger Storage

**Chain-code Services**
- Secure Container
- Secure Registry

**Event Hub**

Open Source Code: Blockchain built from the ground up for business; Permission | Privacy | Confidential | Auditable
Smart contract

- Verifiable, signed
- Encoded in programming language
- Example:
  - Defines contractual conditions under which a bond transfer occurs

Business rules associated with the transaction
Privacy

- Participants need:
  - Appropriate confidentiality between subsets of participants
  - Identity not linked to a transaction
- Transactions need to be authenticated
- Cryptography central to these processes
Trust

- Participants endorse transactions
  - Business network decides who will endorse transactions
  - Endorsed transactions are added to the ledger with appropriate confidentiality
- Assets have a verifiable audit trail
  - Transactions cannot be modified, inserted or deleted
- Achieved through consensus, provenance, immutability and finality
Hyperledger: A Linux Foundation project

- A collaborative effort created to advance cross-industry blockchain technologies for business
- Announced December 2015, now over 150 members
- Open source, open standards, open governance
- Five frameworks and three tools projects
- IBM is a premier member of Hyperledger
Hyperledger Fabric: Distributed ledger platform

• An implementation of blockchain technology that is a foundation for developing blockchain applications

• Emphasis on ledger, smart contracts, consensus, confidentiality, resiliency and scalability.

• V1.0 released July 2017
  – 159 developers from 27 organizations
  – IBM is one contributor of code, IP and development effort to Hyperledger Fabric

http://hyperledger-fabric.readthedocs.io/
Hyperledger Composer: Accelerating Time to Value

https://hyperledger.github.io/composer/

• A suite of high level application abstractions for business networks
• Emphasis on business-centric vocabulary for quick solution creation
• Reduce risk, and increase understanding and flexibility

• Features
  – Model your business networks, test and expose via APIs
  – Applications invoke transactions to interact with business network
  – Integrate existing systems of record
• Fully open and part of Linux Foundation Hyperledger
• Try it in your web browser now: http://composer-playground.mybluemix.net/
Examen final:

- What is Blockchain?
- Why is it relevant for our business?
- How can IBM help us apply blockchain?
Making blockchain real for business with over 400 engagements and multiple active networks

<table>
<thead>
<tr>
<th>Trade Finance</th>
<th>Pre and Post Trade</th>
<th>Complex Risk Coverage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Digital Trade Chain</td>
<td>DTCC</td>
<td>AIG</td>
</tr>
<tr>
<td>NATIXIS</td>
<td>CLS</td>
<td>Standard Chartered</td>
</tr>
<tr>
<td>TRAFIGURA</td>
<td>Bolsa Comercio SANTANDO</td>
<td></td>
</tr>
<tr>
<td>Mizhu</td>
<td>JPX</td>
<td></td>
</tr>
<tr>
<td>SECURE KEY</td>
<td>NORTHERN TRUST</td>
<td></td>
</tr>
<tr>
<td>DIACC</td>
<td>BORSA ITALIANA</td>
<td></td>
</tr>
<tr>
<td>Crédit Mutuel ARKEA</td>
<td>UnionPay</td>
<td></td>
</tr>
<tr>
<td>SBI証券</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Identity/ Know your customer (KYC)</td>
<td>Unlisted Securities/ Private Equity Funds</td>
<td>Loyalty Program</td>
</tr>
<tr>
<td>SECURE KEY</td>
<td>NORTHERN TRUST</td>
<td>UnionPay</td>
</tr>
<tr>
<td>DIACC</td>
<td>BORSA ITALIANA</td>
<td></td>
</tr>
<tr>
<td>Crédit Mutuel ARKEA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SBI証券</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Medicated Health Data Exchange</td>
<td>Fraud/ Compliance Registry</td>
<td>Distributed Energy/ Carbon Credit</td>
</tr>
<tr>
<td>FDA</td>
<td>SMART DUBAI</td>
<td>TENNET</td>
</tr>
<tr>
<td>جبلي الذكية</td>
<td></td>
<td>ENERGY BLOCKCHAIN LABS</td>
</tr>
<tr>
<td>Supply Chain</td>
<td>Food Safety</td>
<td>Provenance/ Traceability</td>
</tr>
<tr>
<td>PSA</td>
<td>everledger</td>
<td></td>
</tr>
<tr>
<td>MERSK</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Walmart</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Driscoll's</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tyson</td>
<td></td>
<td></td>
</tr>
<tr>
<td>McLane</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dole</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Kroger</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Nestlé</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Golden State Fruits</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

© 2017 IBM Corporation
IBM Engagement Model overview

1. Discuss Blockchain technology
2. Explore customer business model
3. Show Blockchain Application demo

1. Understand Blockchain concepts & elements
2. Hands on with Blockchain on Bluemix
3. Standard demo customization

1. Design Thinking workshop to define business challenge
2. Agile iterations incrementally build project functionality
3. Enterprise integration

1. Scale up pilot or Scale out to new projects
2. Business Process Re-engineering
3. Systems Integration

Remote: Digital: Face to face
Face to face
Getting started on your blockchain journey

Learn More About IBM Blockchain

Schedule an IBM Blockchain Workshop

Develop a Blockchain Application

Activate and Grow your Blockchain Network
Thank you

IBM Blockchain

www.ibm.com/blockchain
developer.ibm.com/blockchain
www.hyperledger.org

© Copyright IBM Corporation 2017. All rights reserved. The information contained in these materials is provided for informational purposes only, and is provided AS IS without warranty of any kind, express or implied. Any statement of direction represents IBM's current intent, is subject to change or withdrawal, and represents only goals and objectives. IBM, the IBM logo, and other IBM products and services are trademarks of the International Business Machines Corporation, in the United States, other countries or both. Other company, product, or service names may be trademarks or service marks of others.
Architecture of Hyperledger Fabric v1

Source: https://jira.hyperledger.org/browse/FAB-37
Overview of Hyperledger Fabric v1

- Reflect business processes by specifying who endorses transactions
- Support broader privacy and confidentiality
- Scale the number of participants and transaction throughput
- Eliminate non deterministic transactions
- Enable pluggable data store
- Be able to dynamically upgrade fabric and chaincode
- Remove SPF and enable multiple providers of Membership Services
The Participants in a Blockchain Network

- **Regulator (R)**: Performs oversight.
- **Blockchain Developer (D)**: Creates applications.
- **Blockchain Network Operator (O)**: Operates.
- **Blockchain User (U)**: Accesses security certificates.
- **Certificate Authority**: Ensures security.
- **Traditional Data Sources**: Provides access to data.
- **Traditional Processing Platforms**: Access to logic.

Blockchain transactions flow through the network, connecting these participants and data sources.
<table>
<thead>
<tr>
<th>Role</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blockchain User</td>
<td>the business user, operating in a business network. This role interacts with the Blockchain using an application. They are not aware of the Blockchain.</td>
</tr>
<tr>
<td>Blockchain Regulator</td>
<td>the overall authority in a business network. Specifically, regulators may require broad access to the ledger’s contents.</td>
</tr>
<tr>
<td>Blockchain Developer</td>
<td>the developer of applications and smart contracts that interact with the Blockchain and are used by Blockchain users.</td>
</tr>
<tr>
<td>Blockchain Network Operator</td>
<td>defines, creates, manages and monitors the Blockchain network. Each business in the network has a Blockchain Network operator.</td>
</tr>
<tr>
<td>Certificate Authority</td>
<td>manages the different types of certificates required to run a permissioned Blockchain.</td>
</tr>
<tr>
<td>Traditional Processing Platform</td>
<td>an existing computer system which may be used by the Blockchain to augment processing. This system may also need to initiate requests into the Blockchain.</td>
</tr>
<tr>
<td>Traditional Data Sources</td>
<td>an existing data system which may provide data to influence the behavior of smart contracts.</td>
</tr>
</tbody>
</table>
## Blockchain Components

- **Ledger**: contains the current world state of the ledger and a Blockchain of transaction invocations.
- **Smart Contract**: encapsulates business network transactions in code. Transaction invocations result in gets and sets of ledger state.
- **Consensus Network**: a collection of network data and processing peers forming a Blockchain network. Responsible for maintaining a consistently replicated ledger.
- **Membership**: manages identity and transaction certificates, as well as other aspects of permissioned access.
- **Events**: creates notifications of significant operations on the Blockchain (e.g., a new block), as well as notifications related to smart contracts. Does not include event distribution.
- **Systems Management**: provides the ability to create, change, and monitor Blockchain components.
- **Wallet**: securely manages a user's security credentials.
- **Systems Integration**: responsible for integrating Blockchain bi-directionally with external systems. Not part of Blockchain, but used with it.
Blockchain Applications and the Ledger

1 Client Application in JavaScript using Hyperledger Fabric Client (HFC) SDK
2 Smart Contract implemented using chain code
Blockchain Applications

• Application
  – Focuses on Blockchain user business needs and experience
  – Calls smart contract for interactions with ledger state
  – Can access transaction ledger directly, if required
  – Can process events if required

• Smart Contract
  – Chain code encapsulates business logic. Choice of implementation language
  – Contract developer defines relevant interfaces (e.g. queryOwner, updateOwner …)
  – Different interfaces access ledger state accordingly – consistent read and write provided
  – Each invocation of a smart contract is a “Blockchain transaction”

• Ledger
  – World/Ledger state holds current value of smart contract data
    • e.g. vehicleOwner=Daisy
  – Blockchain holds historic sequence of all chain code transactions
    • e.g. updateOwner(from=John, to=Anthony); updateOwner (from=Anthony, to=Daisy); etc
Client App submits a transaction proposal for Smart Contract A to the Endorsing peer $E_0$

**Note:**
- Endorsement policy: “$E_0$, $E_1$ and $E_2$ must sign”
- $P_3$, is not part of the policy
Endorsing peer $E_0$ executes the tx and (optionally) “anchors it” with respect to the ledger state version numbers.

An “anchor” contains all data read and written by the contract that is to be confirmed by other endorsers.
The client requests further endorsement from $E_1$ and $E_2$ as per the endorsement policy.

The client may decide to suggest an anchor obtained from $E_0$ to $E_1$ and $E_2$. 
The Endorsing peers $E_1$ and $E_2$ sign the result and send the endorsement to the client.
The client formats the transaction and sends it to the **ordering-service** nodes for inclusion in the ledger.
The *ordering-service* delivers the next block in the ledger with the endorsed transaction.
The peers validate the block received from the **ordering-service** and update their ledger and worldstate.
Endorsement Policies

An endorsement policy describes the conditions by which a transaction can be endorsed. A transaction can only be considered valid if it has been endorsed according to the policy.

- Peers maintain a set of endorsement policies
- An endorsement policy is specified on deployment of chaincode
Endorsement Policy Examples

• Endorser set:
  
  \{DVLA, Nissan, Audi, Toyota, Jaguar, BMW, Honda, LeasePlus, TopLease\}

• Example endorsement policies
  
  – New car transaction from only \{DVLA, Nissan\}
  – Change of ownership transaction from \{DVLA, Nissan, LeasePlus\}

• Other examples
  
  – All endorsers validate transaction
  – A subset of endorsers validate transaction using AND / OR
  – A subset of endorsers validate transaction based on a weighting value
Nodes send/receive messages to the ordering-service via channels.

– Enables chaincode privacy
  • Chaincode deployed to certain nodes
– Messages partitioned into separate channels
  • Transactions stored depending on node and channel
– Nodes can connect to one or more channels
The ordering service packages transactions into blocks to be delivered to peers. Communication with the service is via channels.

Different configuration options for the ordering service include:

- **SOLO**
  - Single node for development

- **Kafka / Zookeeper**
  - 1:n nodes providing Crash Fault Tolerance
  - Odd number of nodes recommended

- **SBFT** (future)
  - 1:n nodes providing Byzantine Fault Tolerance
Requests certificates 1xEcert, NxTcert

User A uses Ecert Tcert

invokes SC txn (signed with TkeyA, encrypted with Vkey/TkeyA)

Enrollment certificates (Ecerts) and Transaction certificates (Tcerts) can only be linked by CA and user

Accesses ledger

Application

Blockchain

Membership

Certificate Authority

Application

Smart contract

deployed on peers

Consensus Network

permissioning will change with v1.0
Transaction and Identity Privacy

• Transaction Certificates, Tcerts
  – Disposable certificates, typically used once, requested from Transaction CA
  – Tcert derived from long term identity - Enrollment Certificate, Ecert
  – Only Transaction CA can link Ecert and Tcert

• Permissioned Interactions
  – Consumer shares public Tcert to provider
  – Provider invokes chain code transaction as usual, but
    • Signs with provider’s private Tcert for authentication
    • Encrypts with provider and consumer Tcerts for subsequent access
  – Consumers can subsequently access ledger data using their private key

• Secure chain code
  – CC can also be signed and encrypted, to keep verify and secure contract details
  – Signing is by contract owner/author
  – Encryption ensures only validators can see and execute transaction chain code
Integrating with Existing Systems (DRAFT)

1. Blockchain events

2. System events

3. Call into Blockchain network from existing systems

4. Call out to existing systems - warning

(1) Though technically possible, chaincode should avoid calling out to external services to prevent non-deterministic errors.
Worldstate

• Underlying storage to be made “pluggable”, so network operators can choose which implementation to use.
• Default embedded key/value implementation will use LevelDB.
• First external database Apache CouchDB:
  – NoSQL database that stores JSON documents.
  – Extensive query capabilities.
Fabric Composer: Accelerating time to value

Open Source development tools from IBM for efficient Blockchain Application development

- Business Network Definition
- Rapid Application Development
- Existing System Integration

Saves time
Reduces risk
Increases flexibility
Increases understanding
Fabric Composer: A Complete Solution Framework Toolset

- CLI utilities
- Data model
- Business-oriented programming language
- Composer playground
- Development environment
- Integration and Analytics
- Web UI
- Blockchain
Key development concepts: Business Network

- Business model defines **participants, assets** and **transactions** (& how they relate)
  - Includes expressive syntax such as arrays, enumerations and references
  - Access Control Lists define rules for sharing and privacy

- **Transaction processors** implement additional business requirements
  - Uses standard Javascript for ease of development and portability

- In Fabric Composer, a business network is defined as
  - A model definition file
  - A set of Javascript processors
  - Packaged into Business Network Archives for ease of deployment
Composer: Development Playground

- Fabric Composer provides a standalone web-based development environment for creating, deploying and updating assets
Key Development Concepts: Participants

1. Defining a participant
2. Participant class name
3. Data structures that define the class. ‘o’ indicates has-a relationship
4. Key fields
5. Registry for storage
Key Development Concepts: Assets

1. Defining an asset
2. Data structures can include complex types (e.g. Enums, participants) ‘o’ indicates has-a relationship
3. Fields can be optional
Key Development Concepts: Transactions

1. Defining a transaction

```java
transaction RegisterPropertyForSale identified by transactionId {
  o String transactionId
  --> Person seller
  --> LandTitle title
}
```

2. Data structures that define the class

‘--->’ indicates pass by reference
Key Development Concepts: Processors

```javascript
/*
 * Sample transaction processor function.
 */
function onSampleTransaction(sampleTransaction) {
  sampleTransaction.asset.value = sampleTransaction.newValue;
  return getAssetRegistry('org.acme.biznet.SampleAsset')
    .then(function (assetRegistry) {
      return assetRegistry.update(sampleTransaction.asset);
    });
}
```
Simple Development Lifecycle

1. Install IBM Solutions Framework Tools
2. Define Business Network, Assets and Transactions
3. Implement any transaction processors
4. Generate application
5. Deploy to a server
