

# Blockchain Analytics

6/6/2022

# ÍNDICE

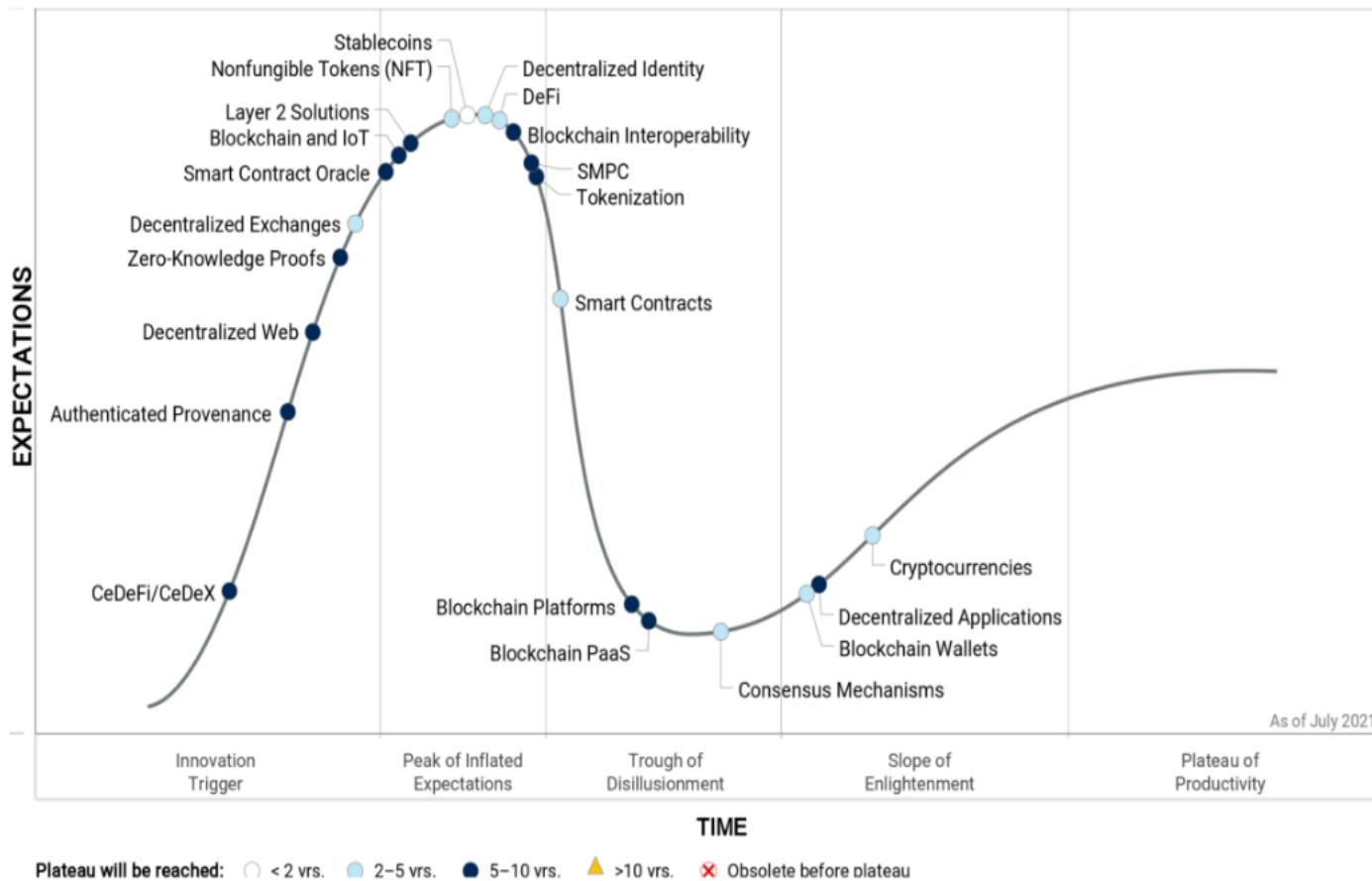
1. Introducción a Blockchain
2. Cómo funciona una Blockchain
3. Taller práctico: desplegando una red de Blockchain



# **INTRODUCCIÓN A BLOCKCHAIN**

# El “hype cycle” de Blockchain

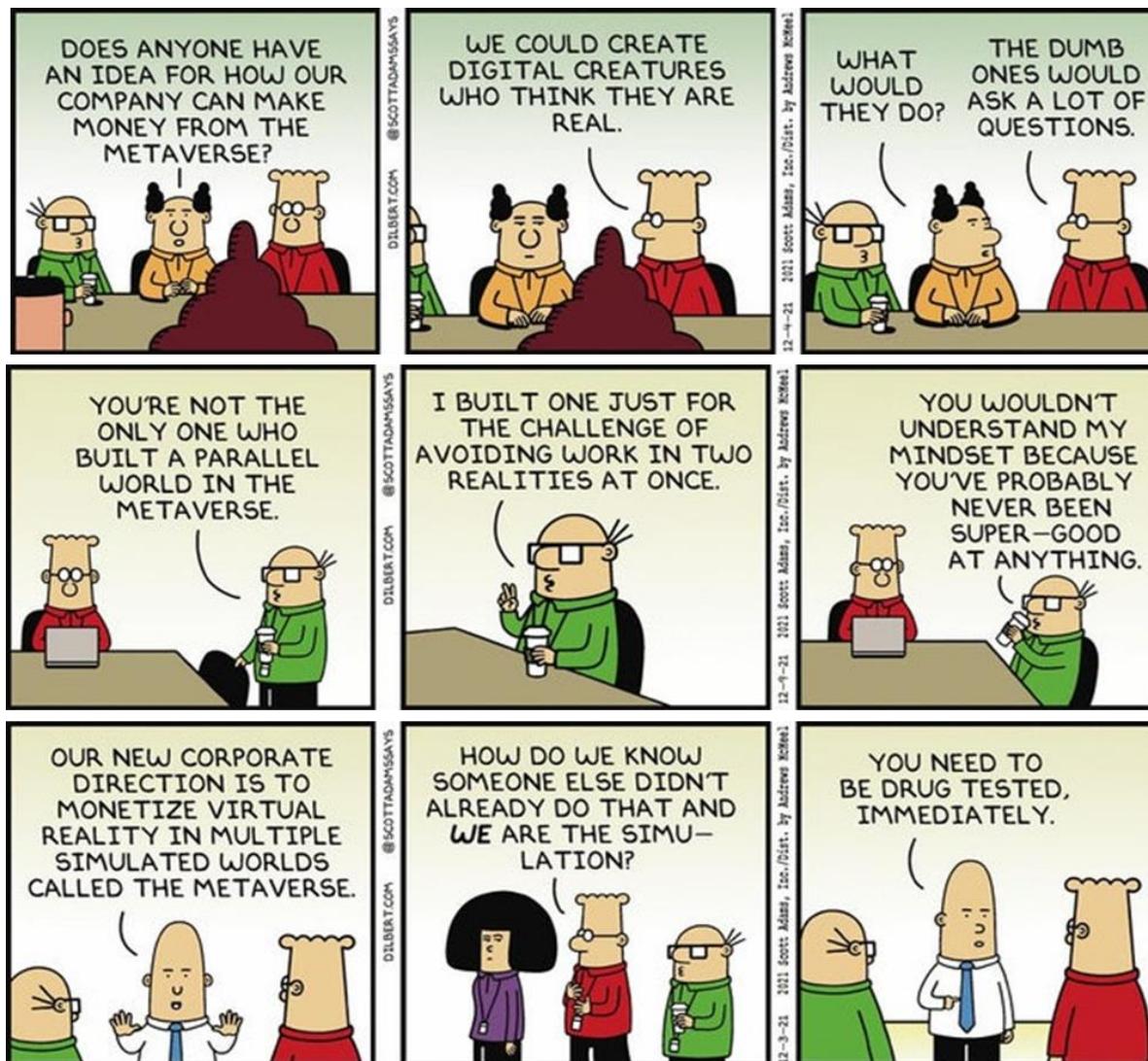
Hype Cycle for Blockchain, 2021



Source: Gartner (July 2021)

747513

# El metaverso, la última frontera



# Expectativas de negocio en los próximos años



"Continuous growth in **blockchain spending** across Europe, from over \$800 million in 2019 to **\$4.9 billion in 2023**"

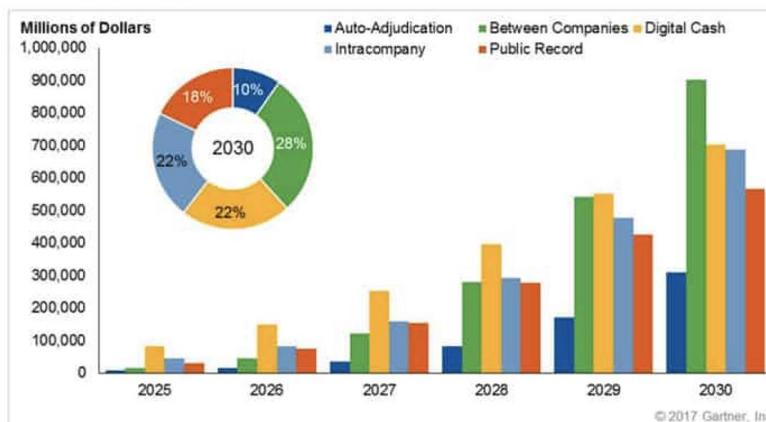


"By 2027, about **10% of the global GDP** will be stored using blockchain"



"**Business value generated by blockchain will grow rapidly, reaching \$176 billion by 2025 and \$3.1 trillion by 2030**"

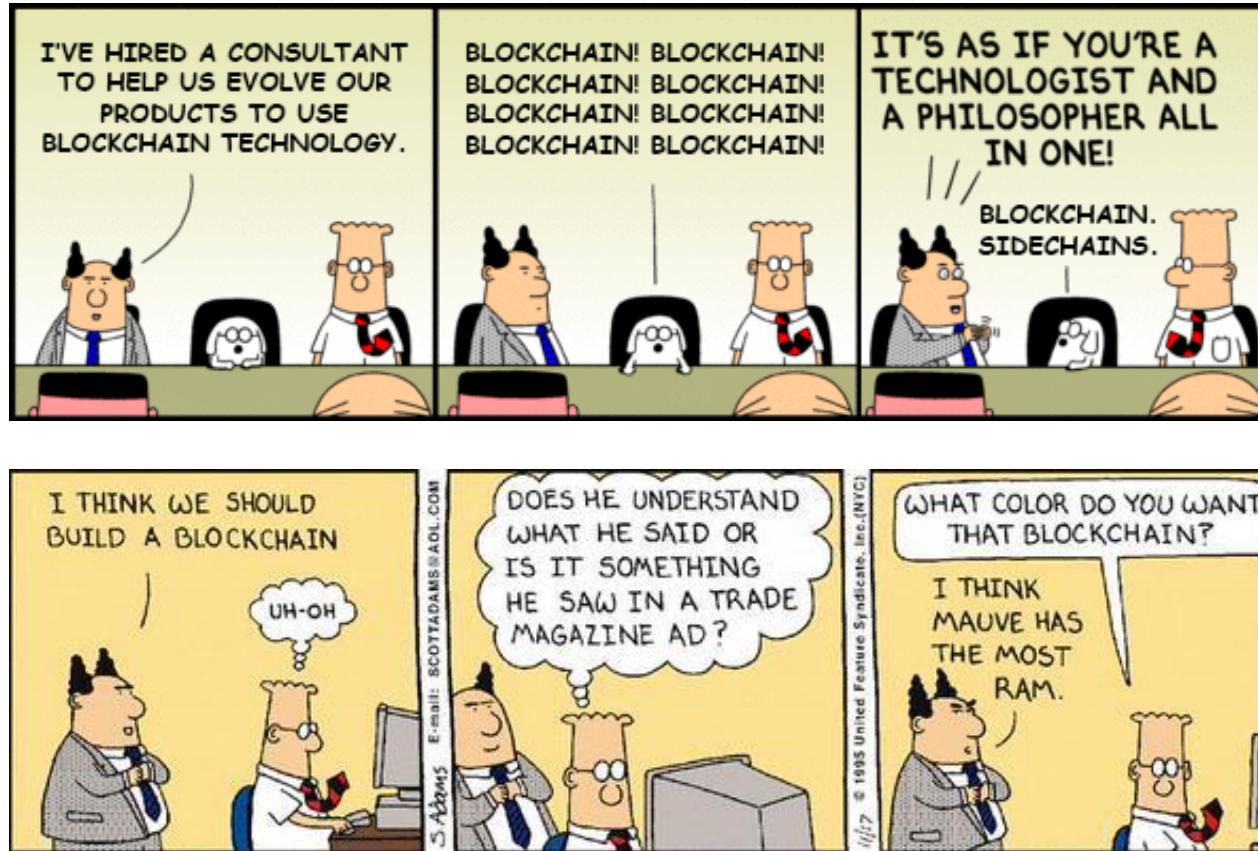
**Business value-add of Blockchain - \$176 billion by 2025, \$3.1 trillion by 2030**



Source: Forecast: Blockchain Business Value, Worldwide, 2017-2030



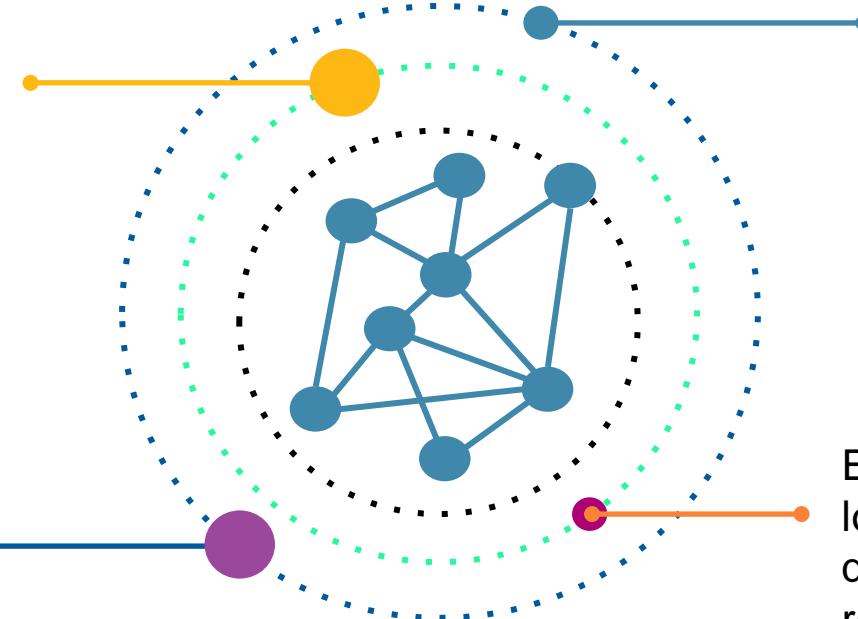
# Una tecnología con muy pocos expertos



# ¿Qué es Blockchain?

Blockchain es un registro descentralizado de información, que no puede modificarse una vez registrada

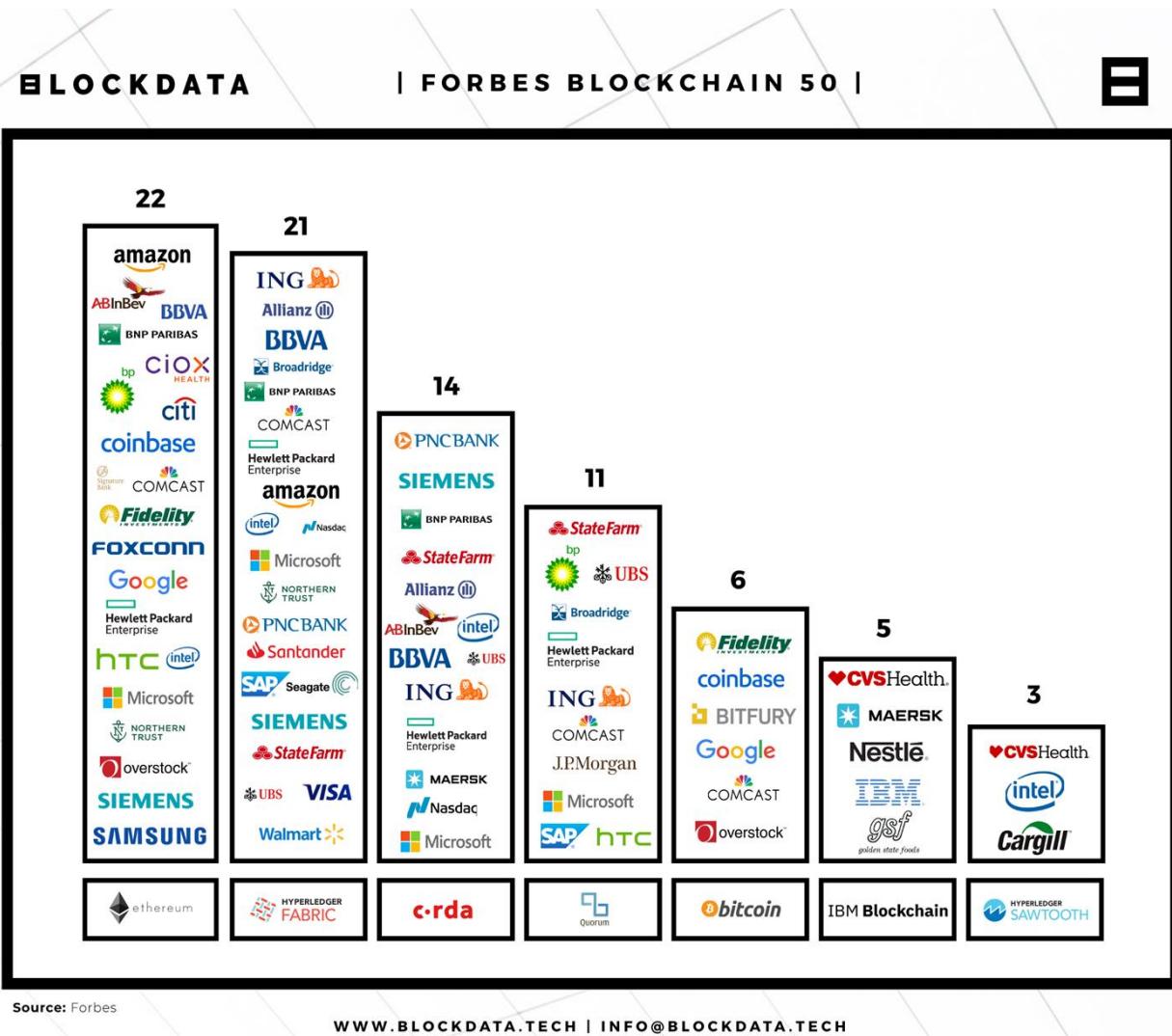
Hay diferentes tecnologías, públicas, privadas y permisionadas



Los nodos de la red alcanzan un consenso sobre la información a registrar en cada momento

Es posible definir lógica de negocio que se ejecuta en la red (Smart contracts)

# ¿Hay más de un Blockchain?



# Blockchain pública vs privada

“Blockchain technology”



private (intra-)

Intranets & IT

Microsoft **IBM ORACLE**

“The Bitcoin Blockchain”



public (inter-)

The Internet

facebook Google amazon

# Smart Contracts (Ethereum)



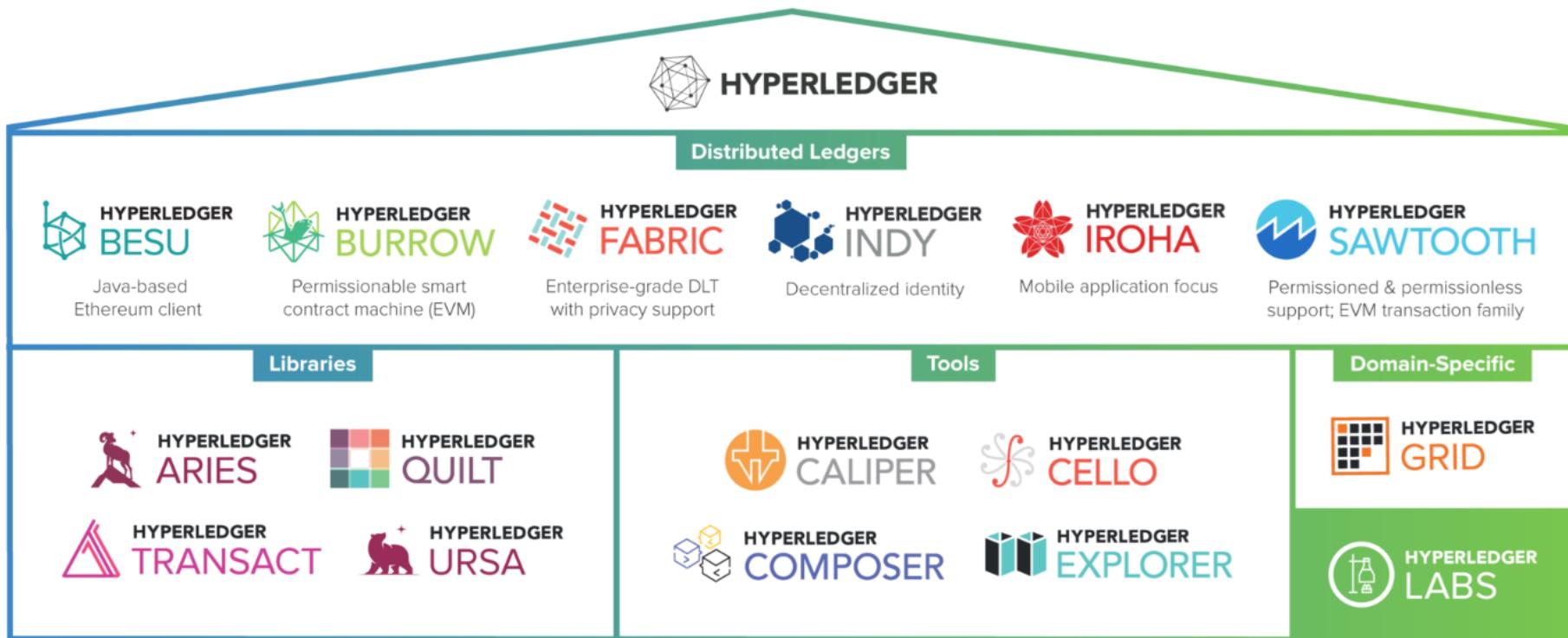
# Quorum



**Quorum:**  
**Ethereum for enterprise**  
**applications**

[jpmorgan.com/quorum](http://jpmorgan.com/quorum)

# Hyperledger



# ¿Usar Blockchain? ¿Cuál?

¿Necesitas una BBDD?

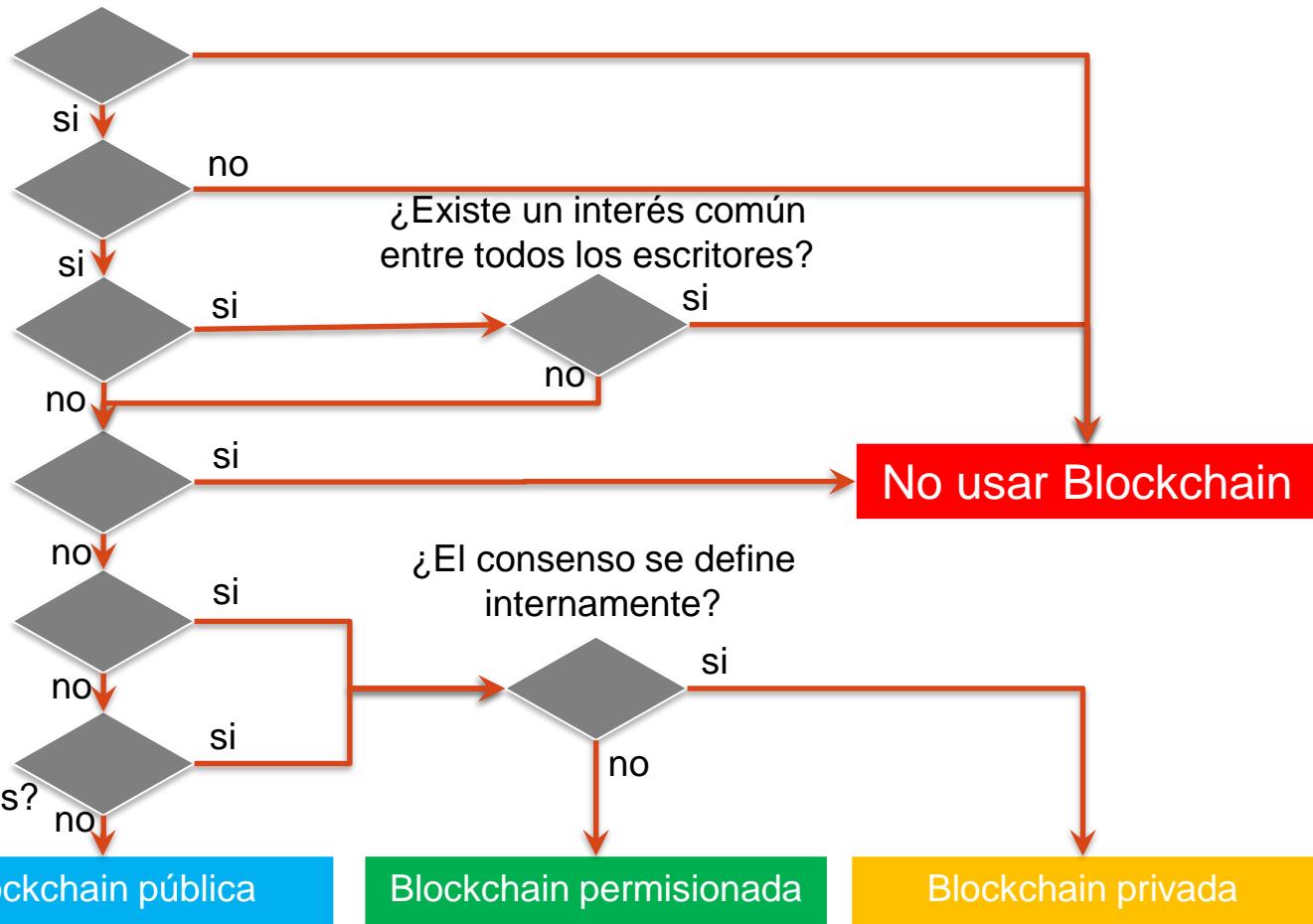
¿Necesitas escritura compartida en la BBDD?

¿Confías en quienes escriben en la BBDD?

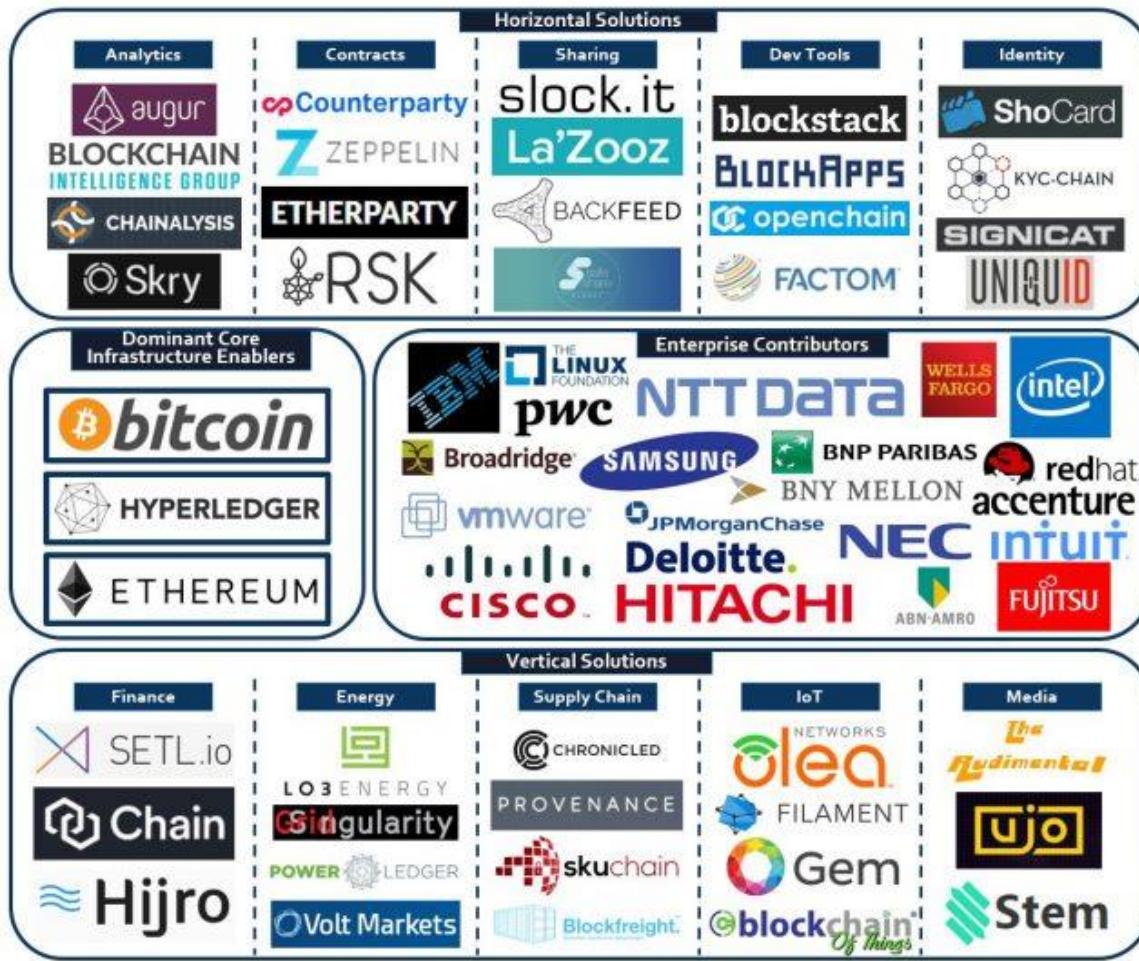
¿Necesitas/confías en utilizar 3as partes?

¿Necesitas controlar la funcionalidad y/o borrar información ya escrita?

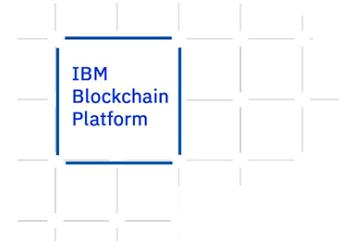
¿Quieres que las transacciones sean privadas?



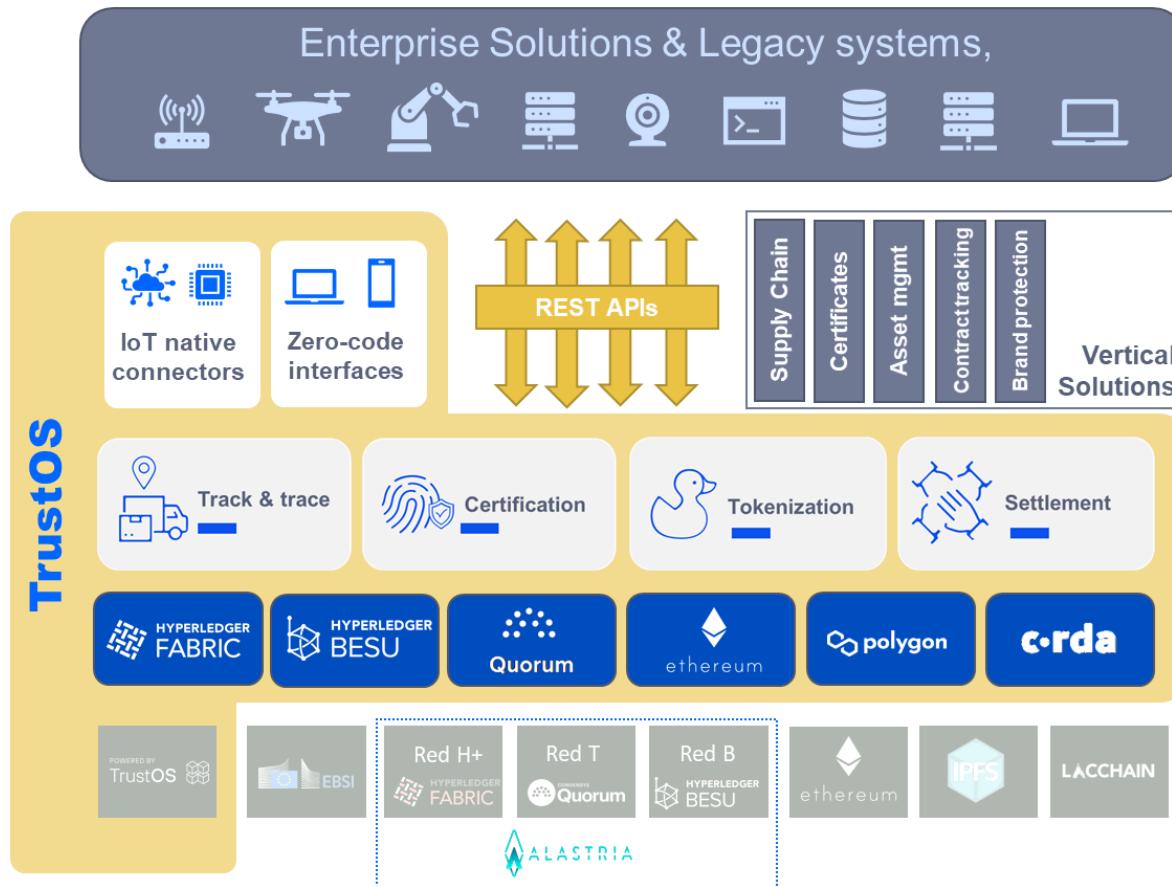
# Blockchain – ecosistema



# Blockchain-as-a-service



# TrustOS: Blockchain rápido y sencillo para empresas

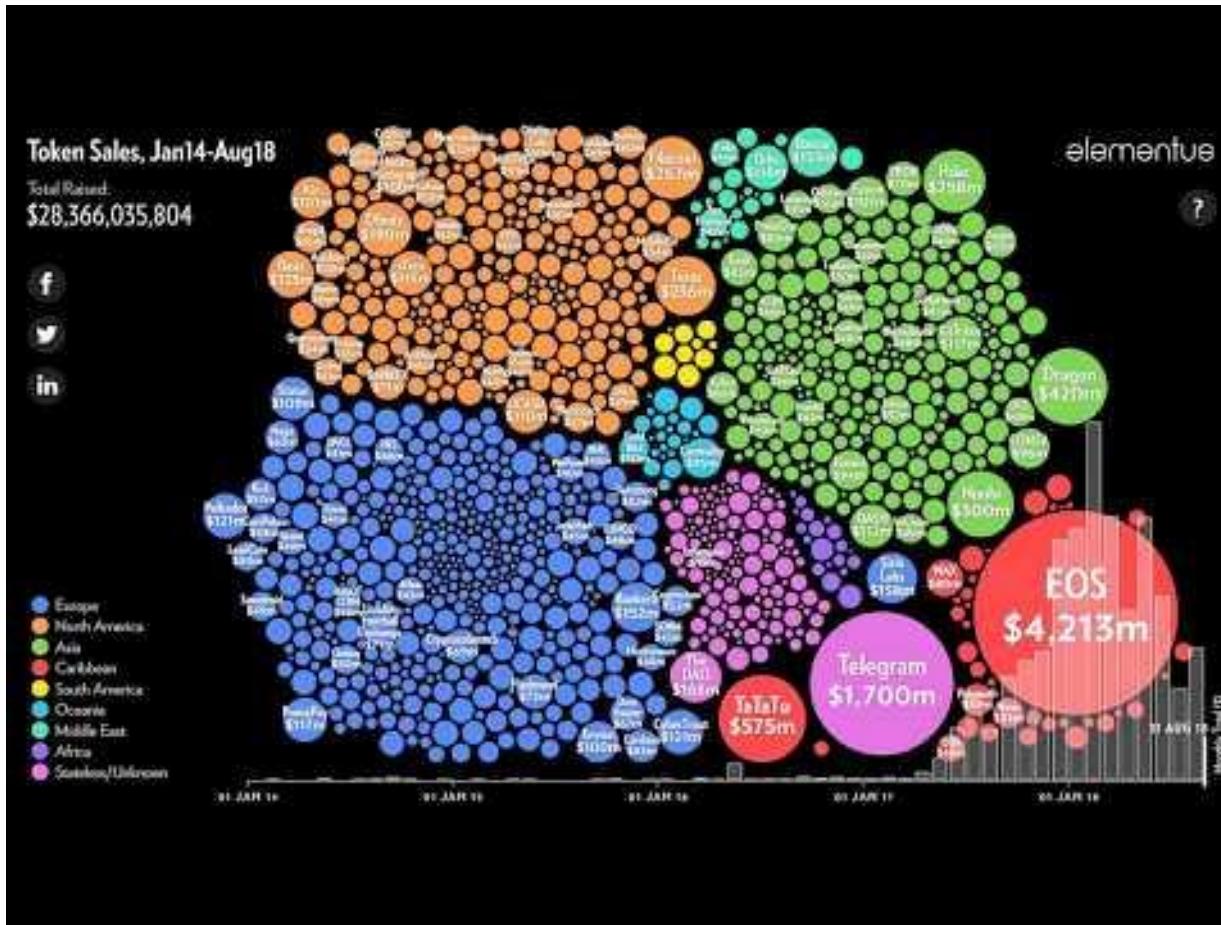


# Casos de uso de Blockchain

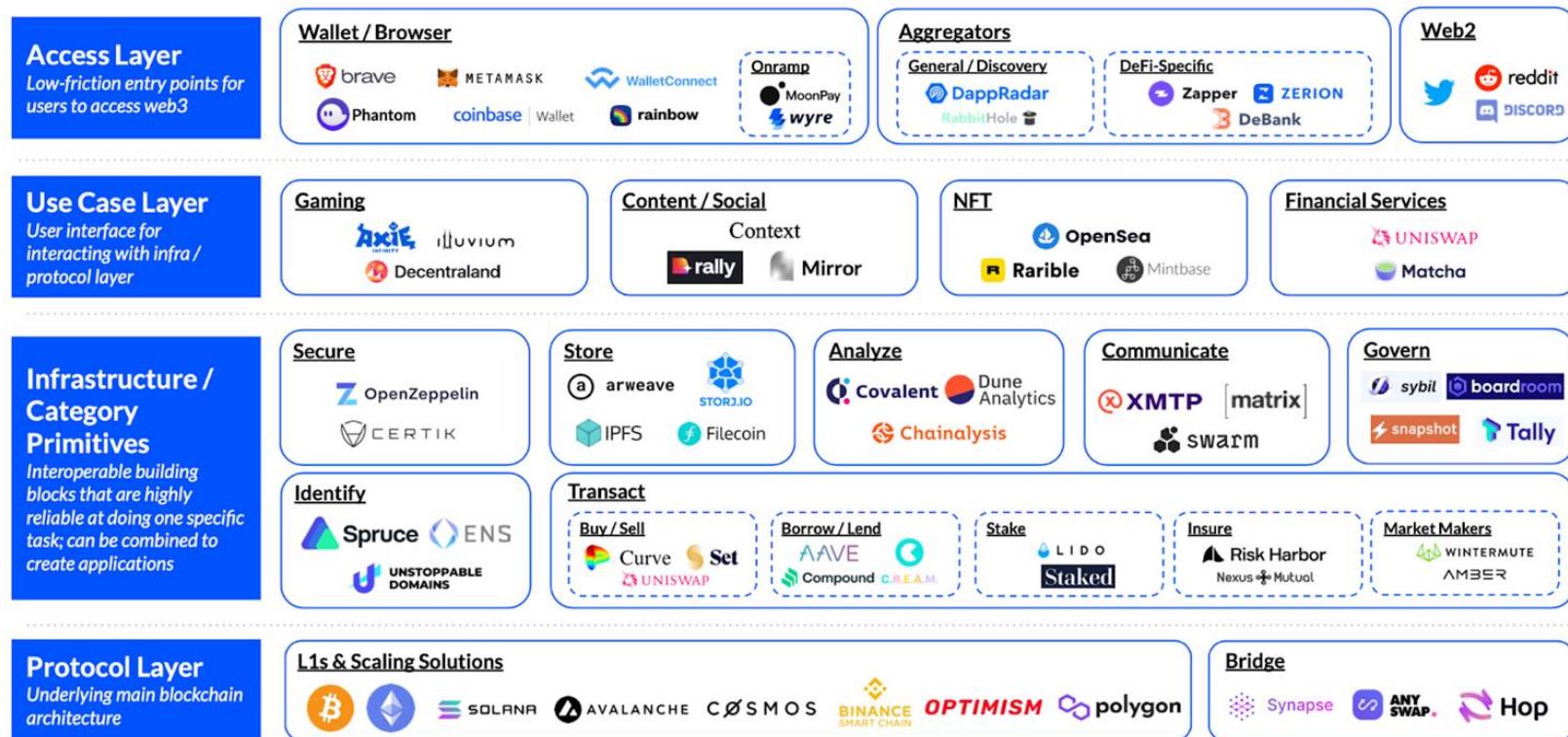


© 2017 Grant Thornton Malta

# Financiación (Initial Coin Offering)



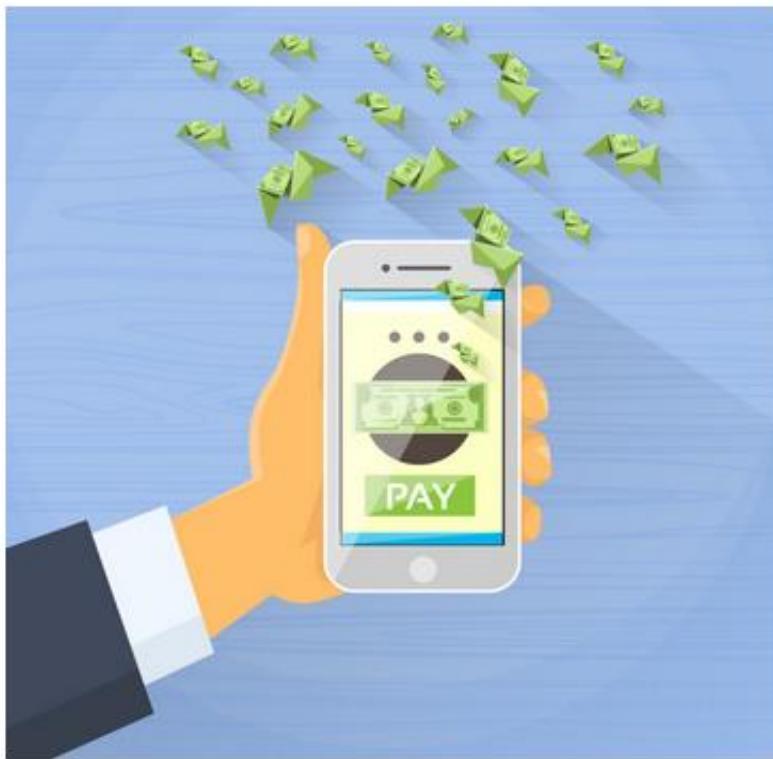
# DeFi (Decentralized Finance)



# Pagos internacionales

## Santander Becomes the First U.K. Bank to Use Ripple for Cross-Border Payments

May 26, 2016 | Monica Long



Today Santander announced that they are the first U.K. bank to introduce Ripple's blockchain technology to facilitate international payments through a new app. They are rolling it out as a staff pilot, with the intention to expand the technology at a later date.

# Bonos corporativos

## Santander Confirms Fiat-backed Token Project on Ethereum Blockchain

4787 Total views 629 Total shares



Santander, a part of the Spanish Santander Group is now using the Ethereum Blockchain to develop a public digital cash system using the Ethereum Blockchain. Santander officials confirmed this project which will make the bank the first to use an existing public Blockchain for issuing digital currency.

# Stablecoins bancarias y CBDCs

**CincoDías** EL PAÍS ECONOMÍA

Compañías Mercados Economía Mi Dinero Fortuna / Cotizaciones f in

ACTUALIDAD ¿Qué laboratorio y qué medicamento facturan más en la farmacia? »

APERTURA El Ibex abre con una caída del 0,13% hasta los 7.114 puntos »

Compañías

PAGOS ONLINE >

## Los cinco grandes bancos se lanzan a por el 'euro digital'

En los próximos días el consejo de administración de Iberpay, compañía española de servicio de pagos en la que participan las principales entidades financieras del país, pondrá fecha para realizar las primeras pruebas para operar con dinero digital, según apuntan varias fuentes financieras.

El documento añade que "este modelo estaría centrado en la emisión de tokens por Iberpay, desde una cuenta común sectorial, cuyos fondos, que serían propiedad de la compañía, actuarían como respaldo del dinero digital emitido, que sería, a su vez, distribuido, en su caso, a los clientes por las entidades financieras".

# Digital euro

We have decided to launch the investigation phase of a digital euro project

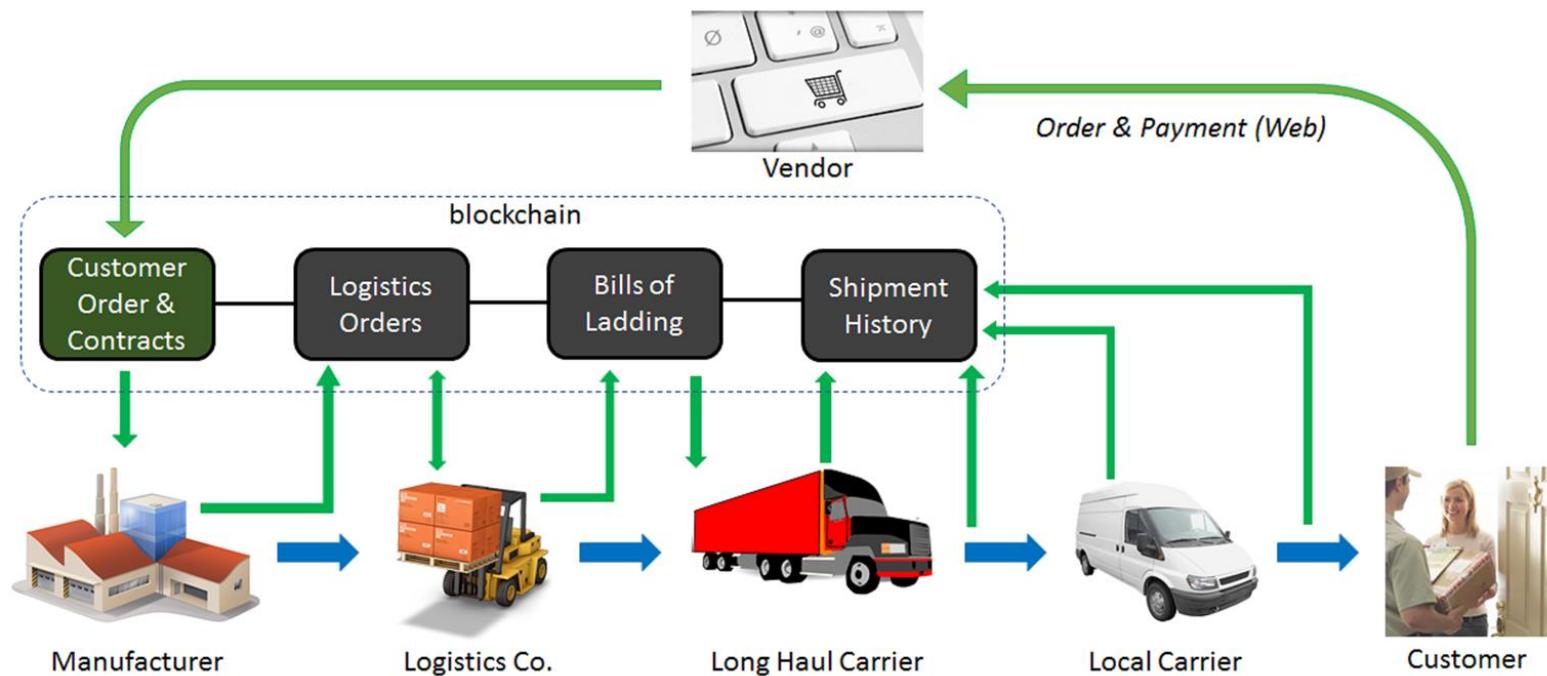
 EUROPEAN CENTRAL BANK  
EUROSYSTEM



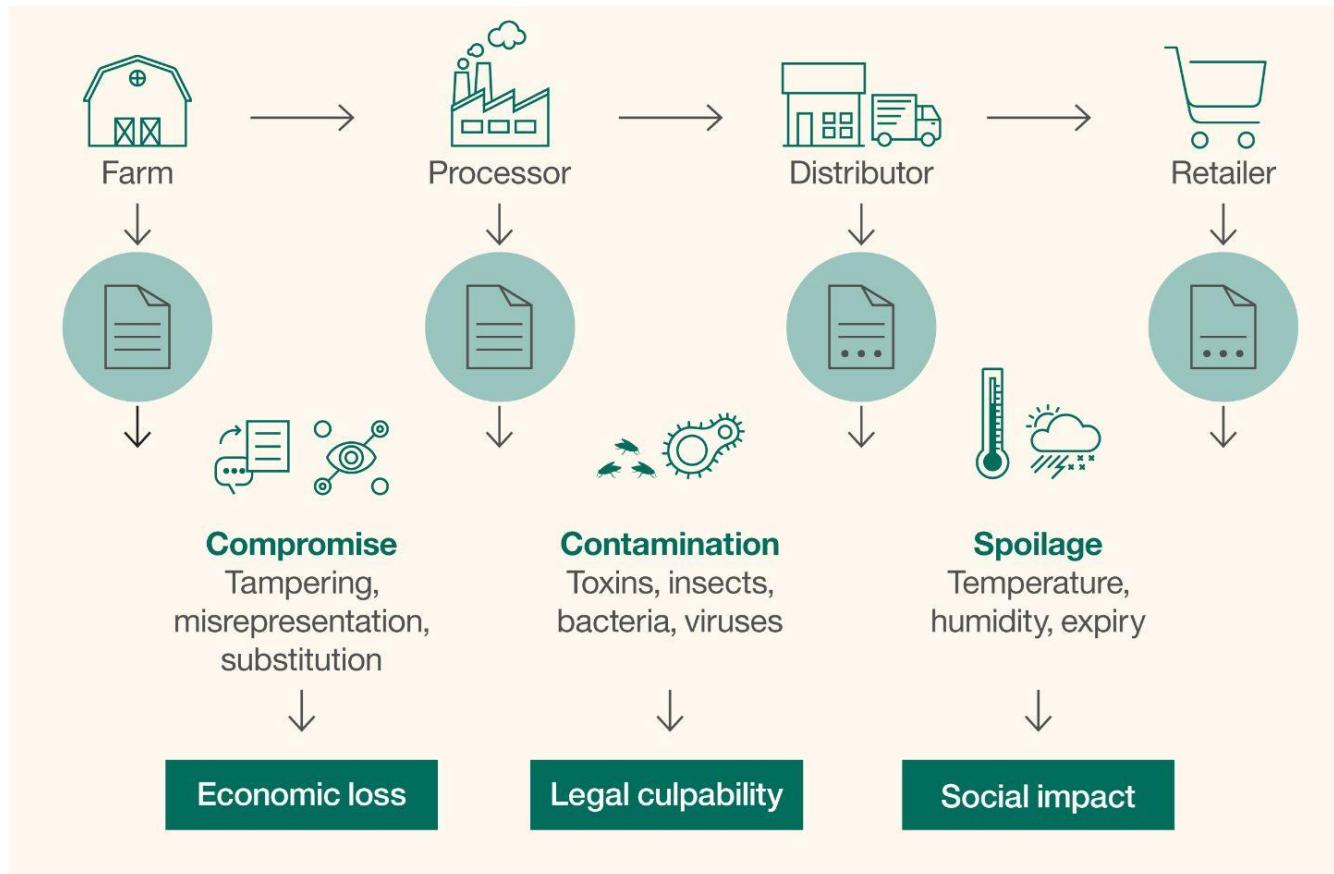
# Stablecoins “privadas”



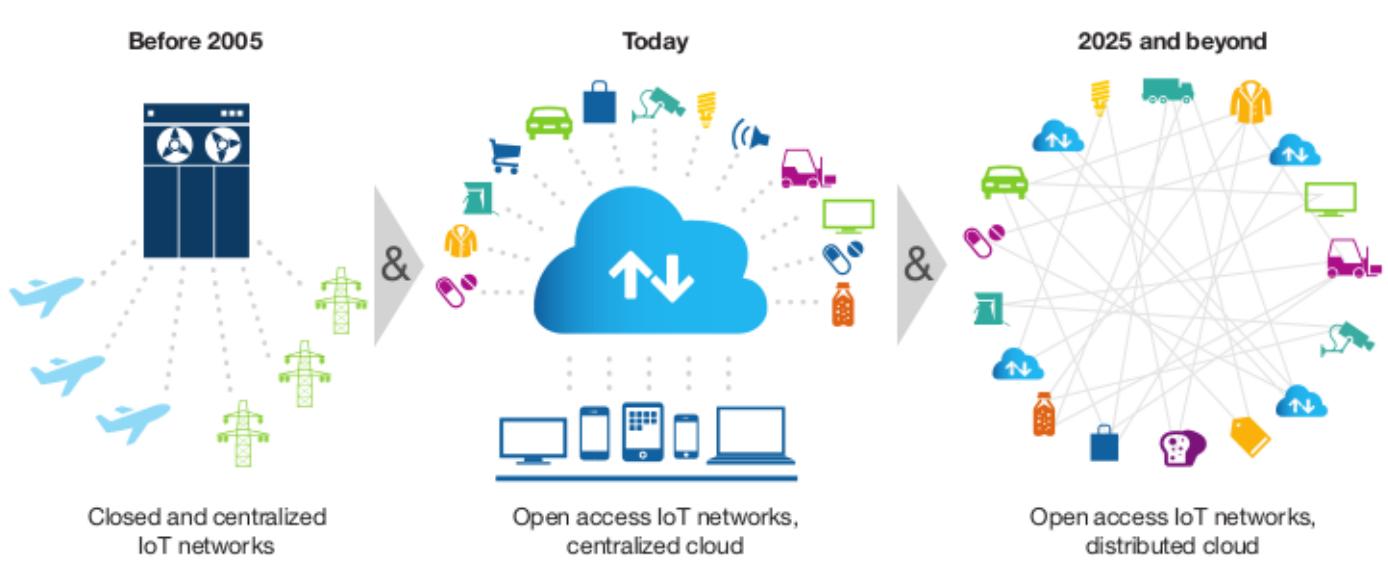
# Logística



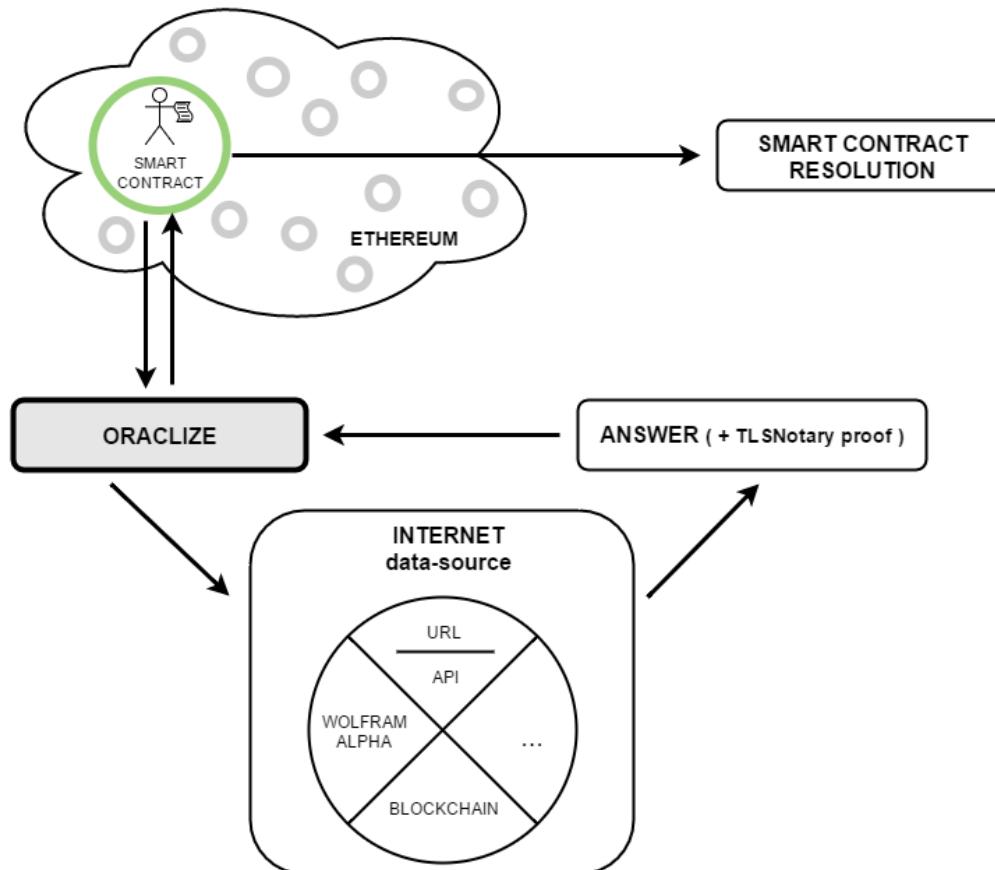
# Trazabilidad



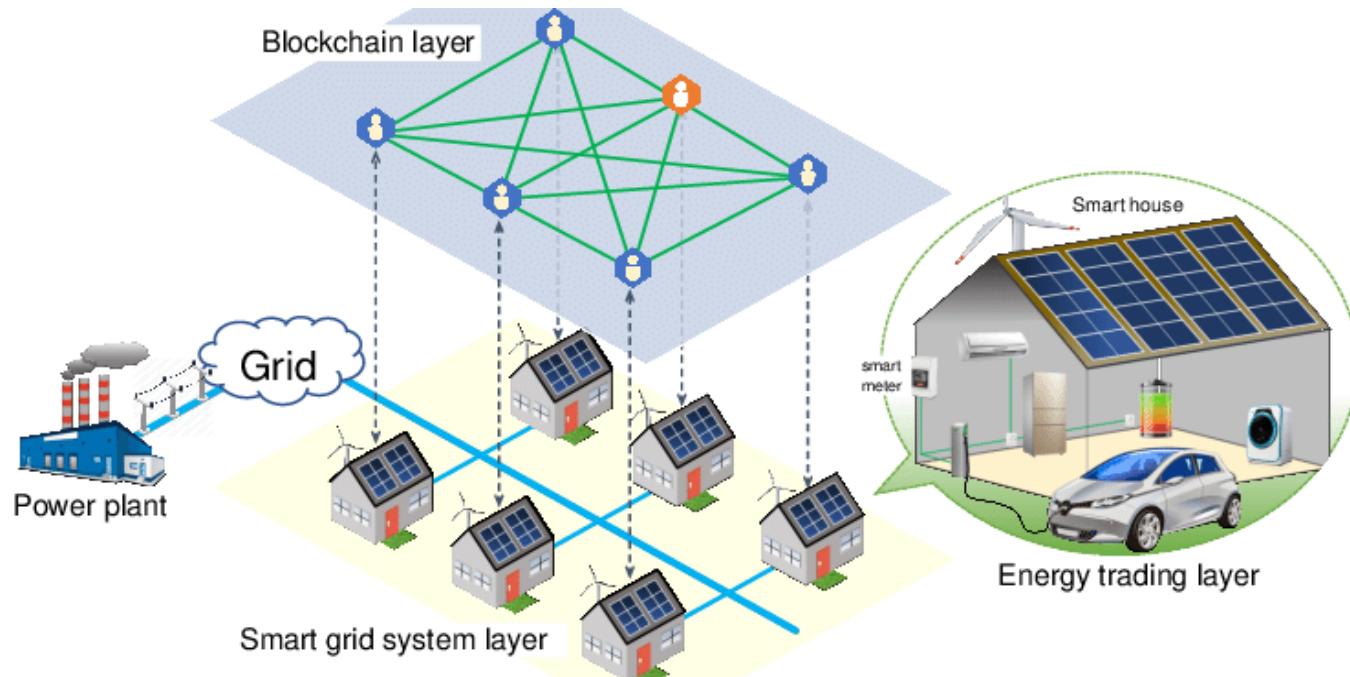
# Internet of Things



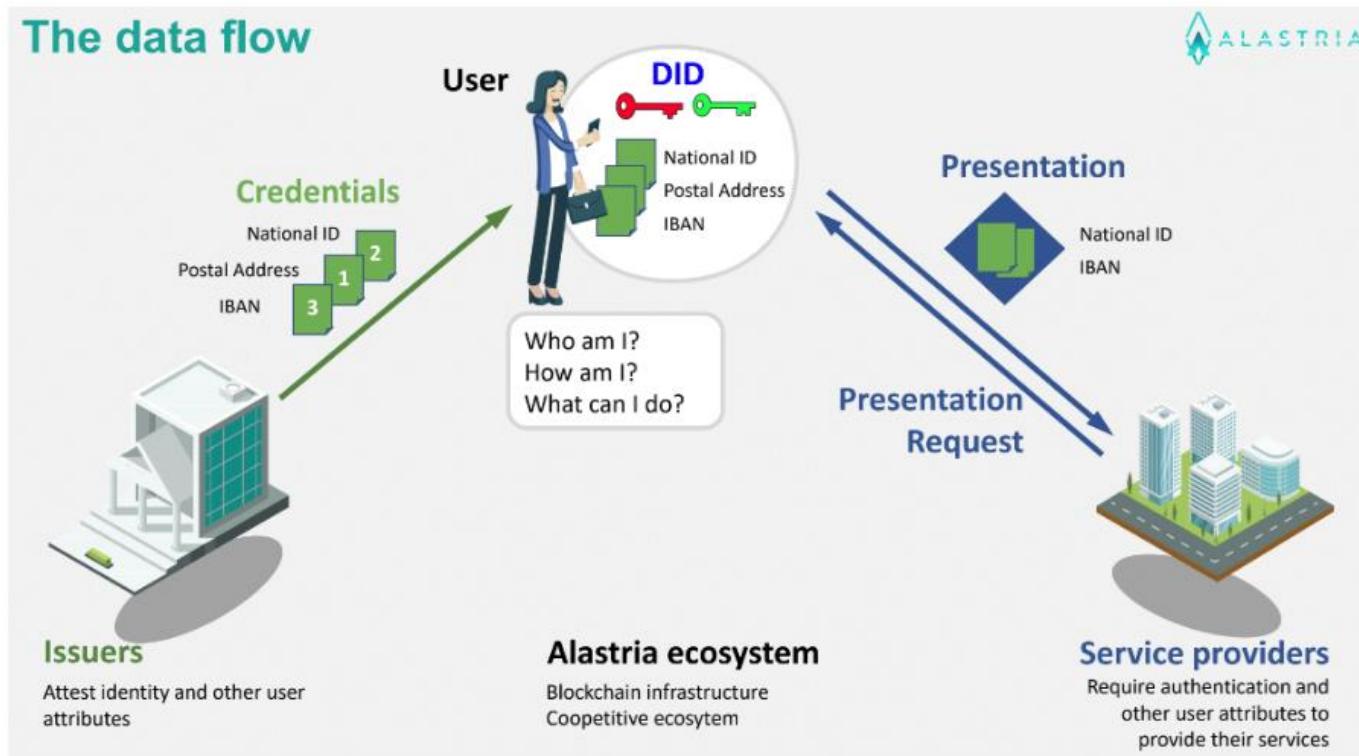
# Seguros inteligentes



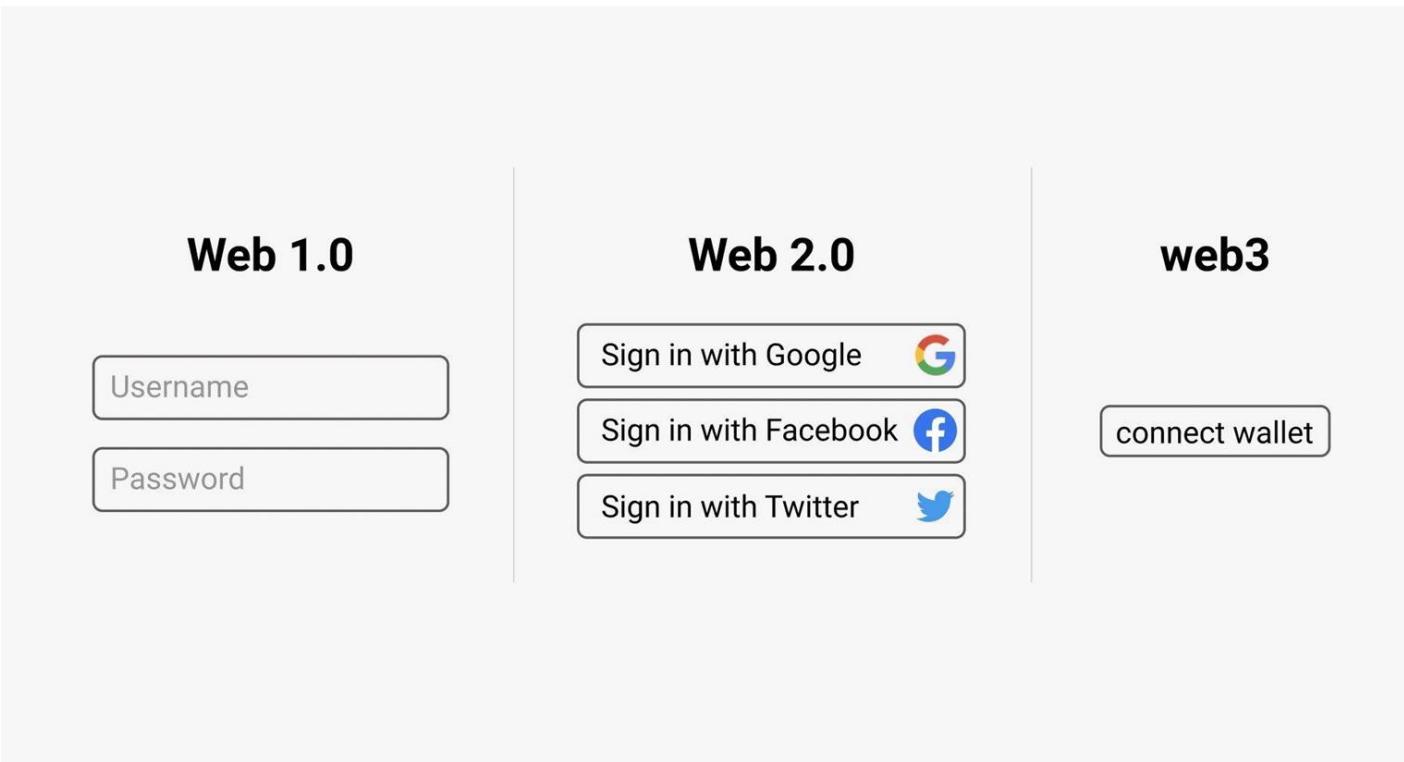
# Energía



# Self Sovereign Identity



# Web3





# CÓMO FUNCIONA BLOCKCHAIN

# Bitcoin, el origen de Blockchain

Satoshi Nakamoto [satoshi at vistomail.com](mailto:satoshi@vistomail.com)  
Fri Oct 31 14:10:00 EDT 2008

- Previous message: [Fw: SHA-3 lounge](#)
- Messages sorted by: [\[ date \]](#) [\[ thread \]](#) [\[ subject \]](#) [\[ author \]](#)

---

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

Abstract. A purely peer-to-peer version of electronic cash would allow online payments to be sent directly from one party to another without the burdens of going through a financial institution. Digital signatures provide part of the solution, but the main benefits are lost if a trusted party is still required to prevent double-spending. We propose a solution to the double-spending problem using a peer-to-peer network. The network timestamps transactions by hashing them into an ongoing chain of hash-based proof-of-work, forming a record that cannot be changed without redoing the proof-of-work. The longest chain not only serves as proof of the sequence of events witnessed, but proof that it came from the largest pool of CPU power. As long as honest nodes control the most CPU power on the network, they can generate the longest chain and outpace any attackers. The network itself requires minimal structure. Messages are broadcasted on a best effort basis, and nodes can leave and rejoin the network at will, accepting the longest proof-of-work chain as proof of what happened while they were gone.

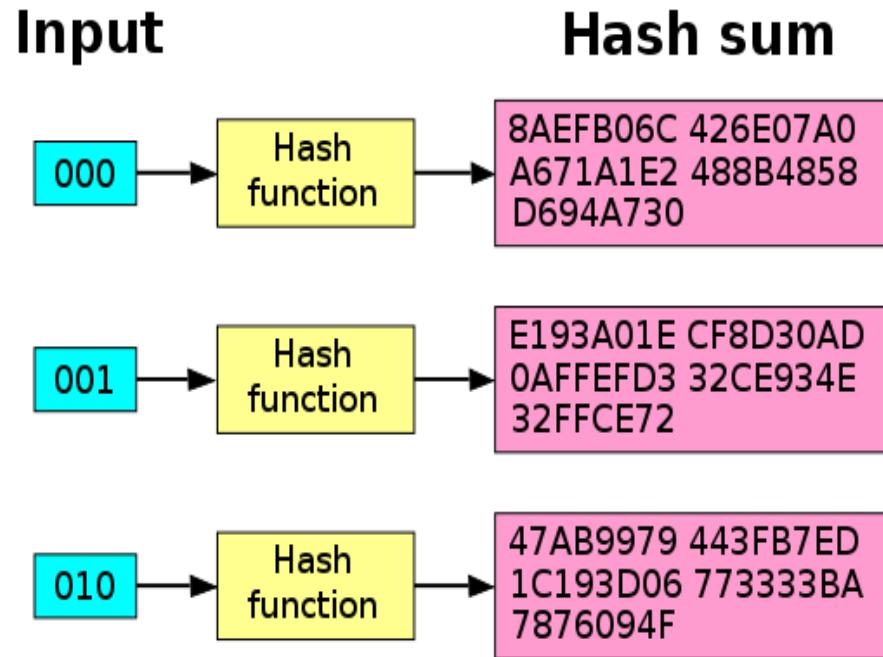
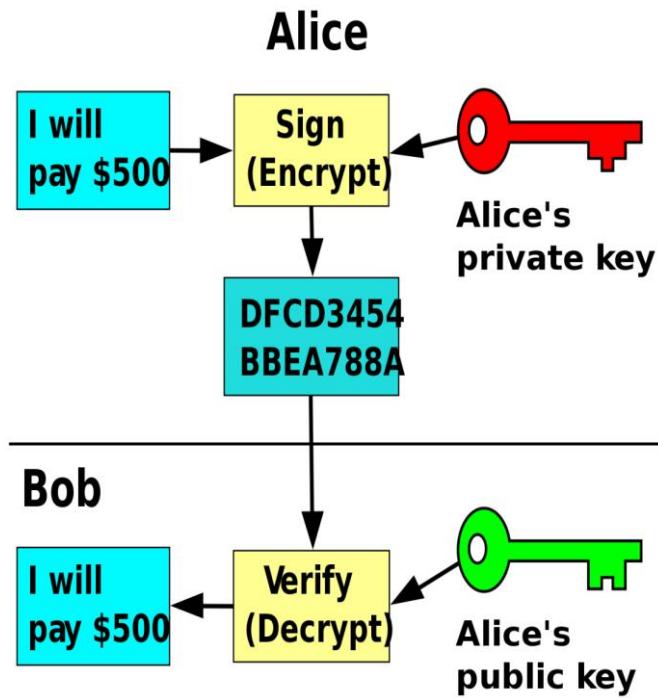
Full paper at:  
<http://www.bitcoin.org/bitcoin.pdf>

Satoshi Nakamoto

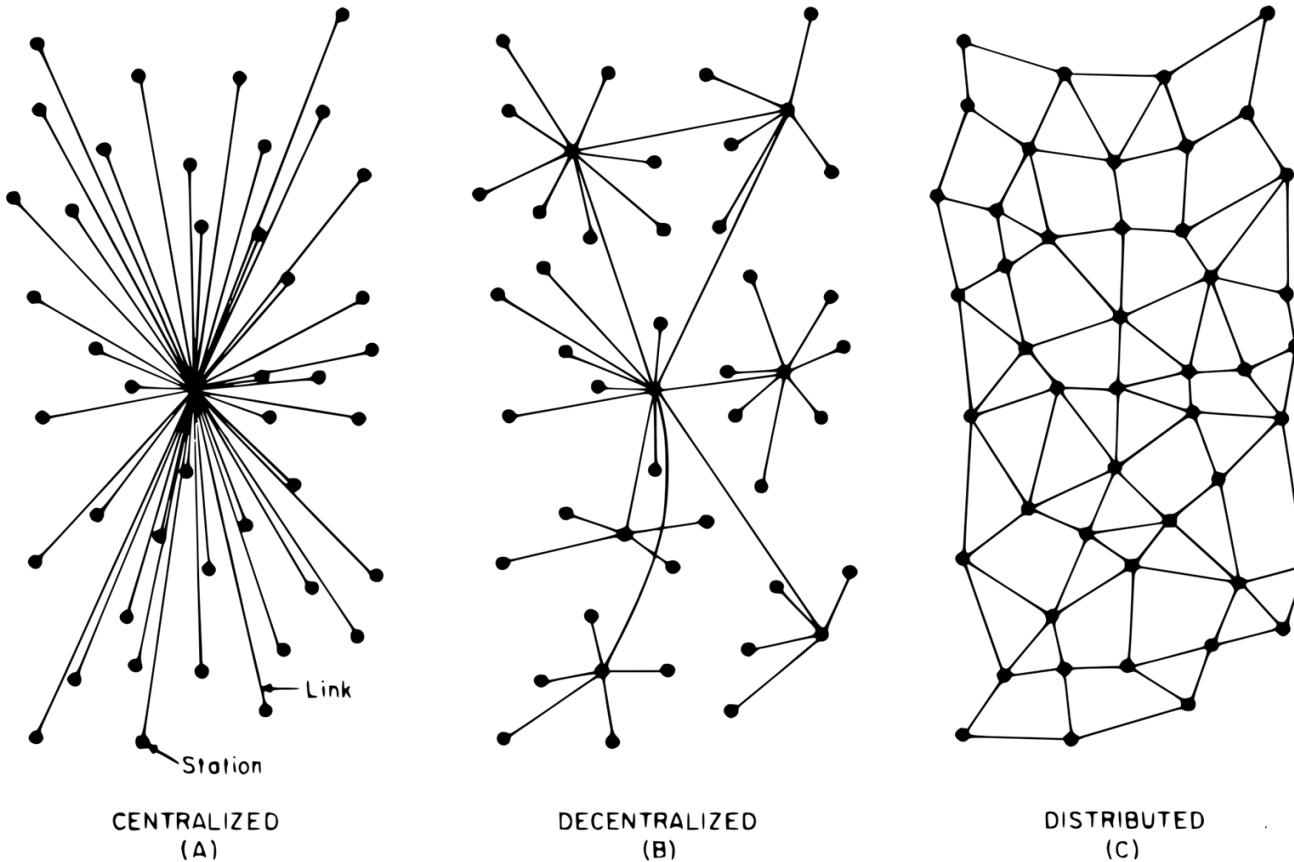
---

The Cryptography Mailing List

# Fundamentos: criptografía



# Fundamentos: redes distribuídas (P2P)



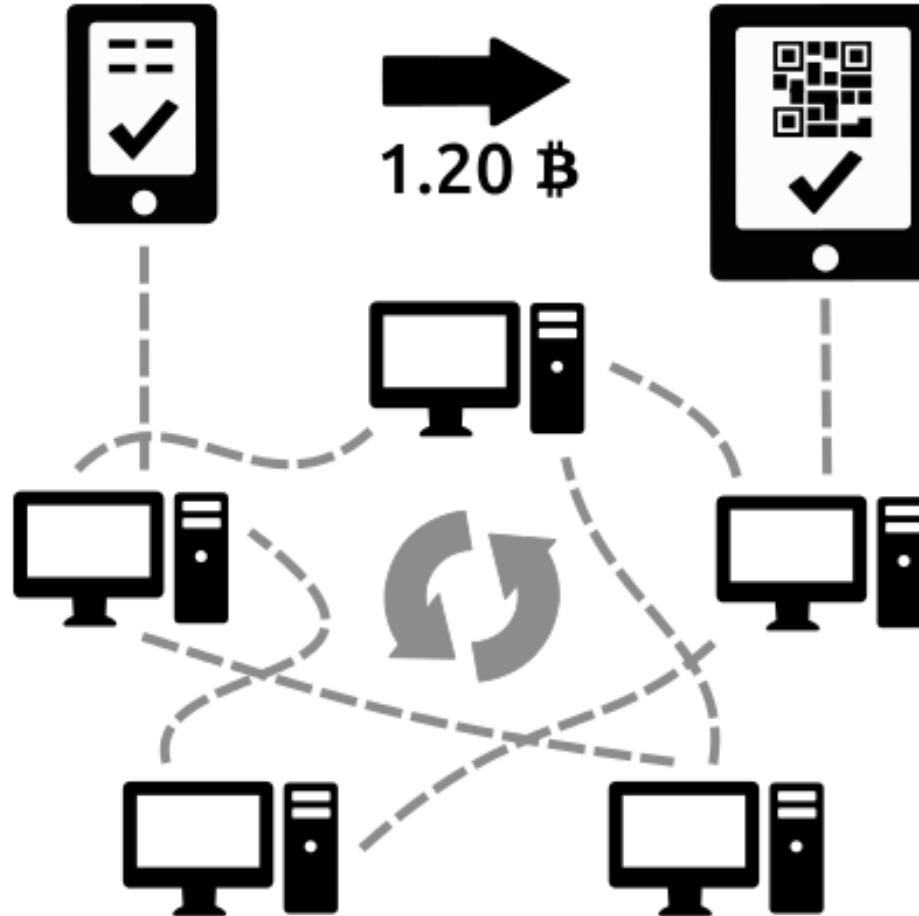
# Fundamentos: consenso (los generales bizantinos)



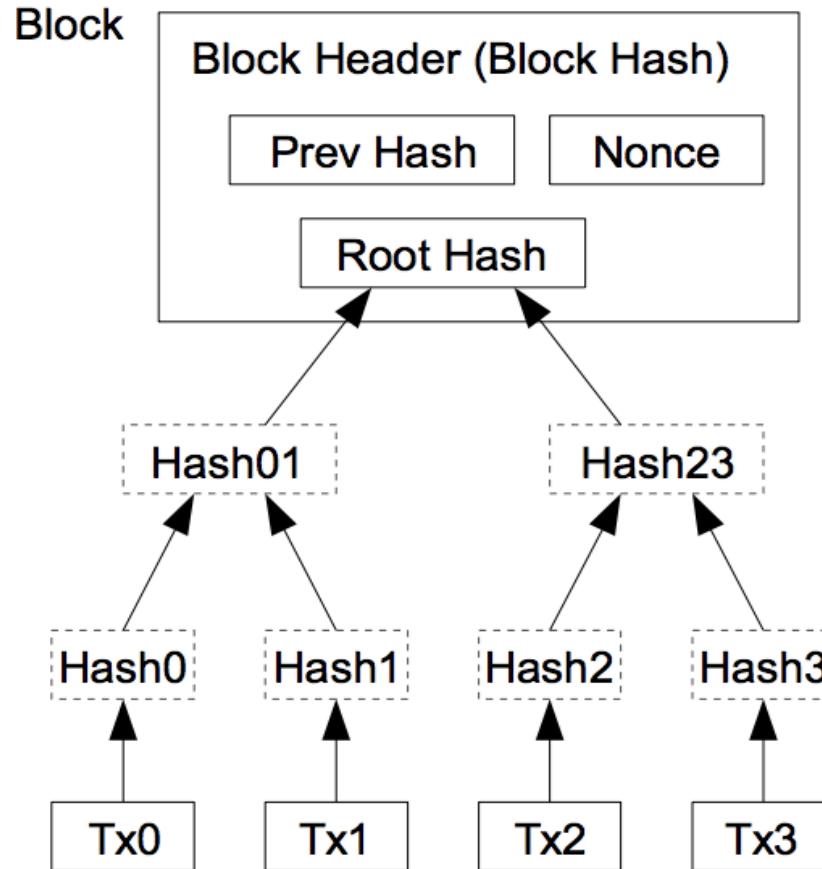
# Fundamentos: incentivos (teoría de juegos)



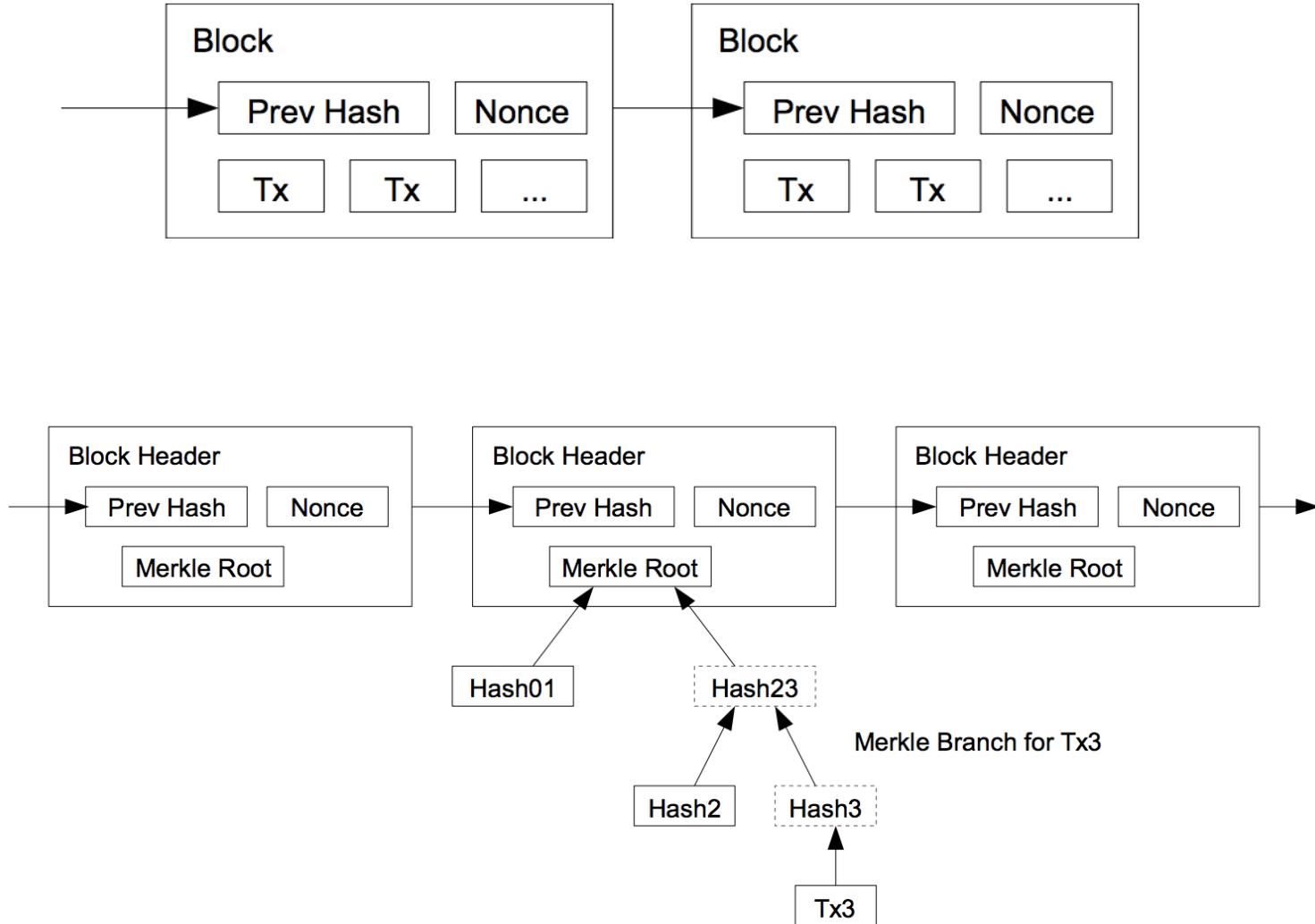
# Cómo funciona Bitcoin



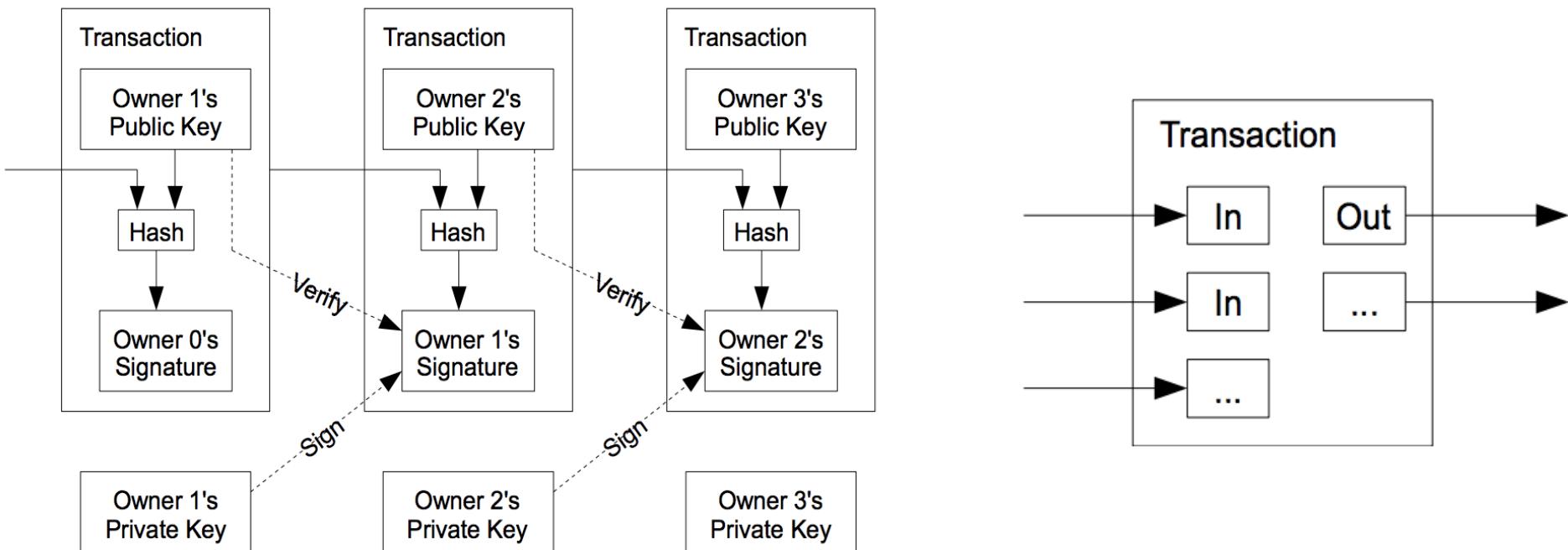
# Cómo funciona Bitcoin: bloques



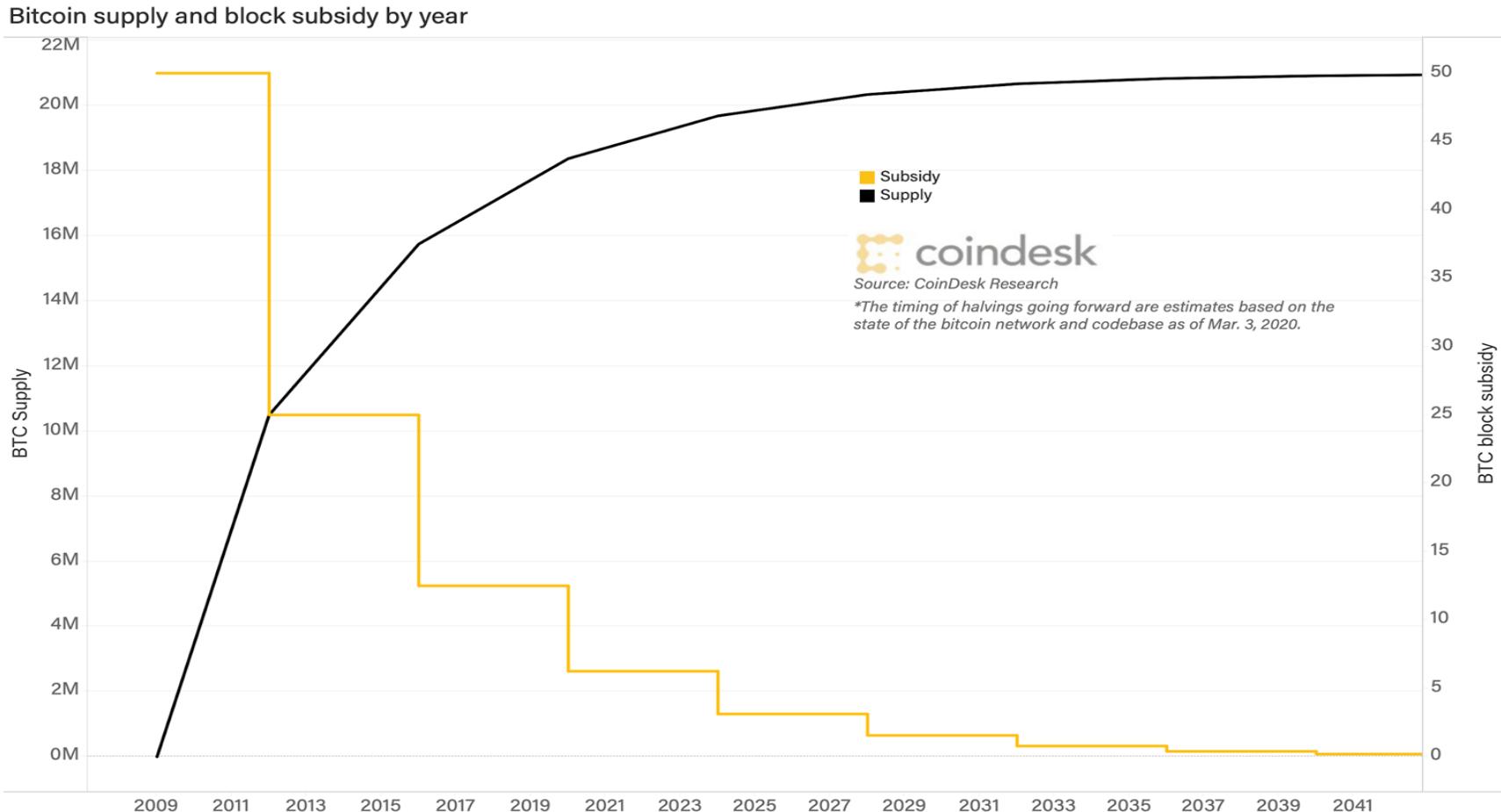
# Cómo funciona Bitcoin: hash y cadena de bloques



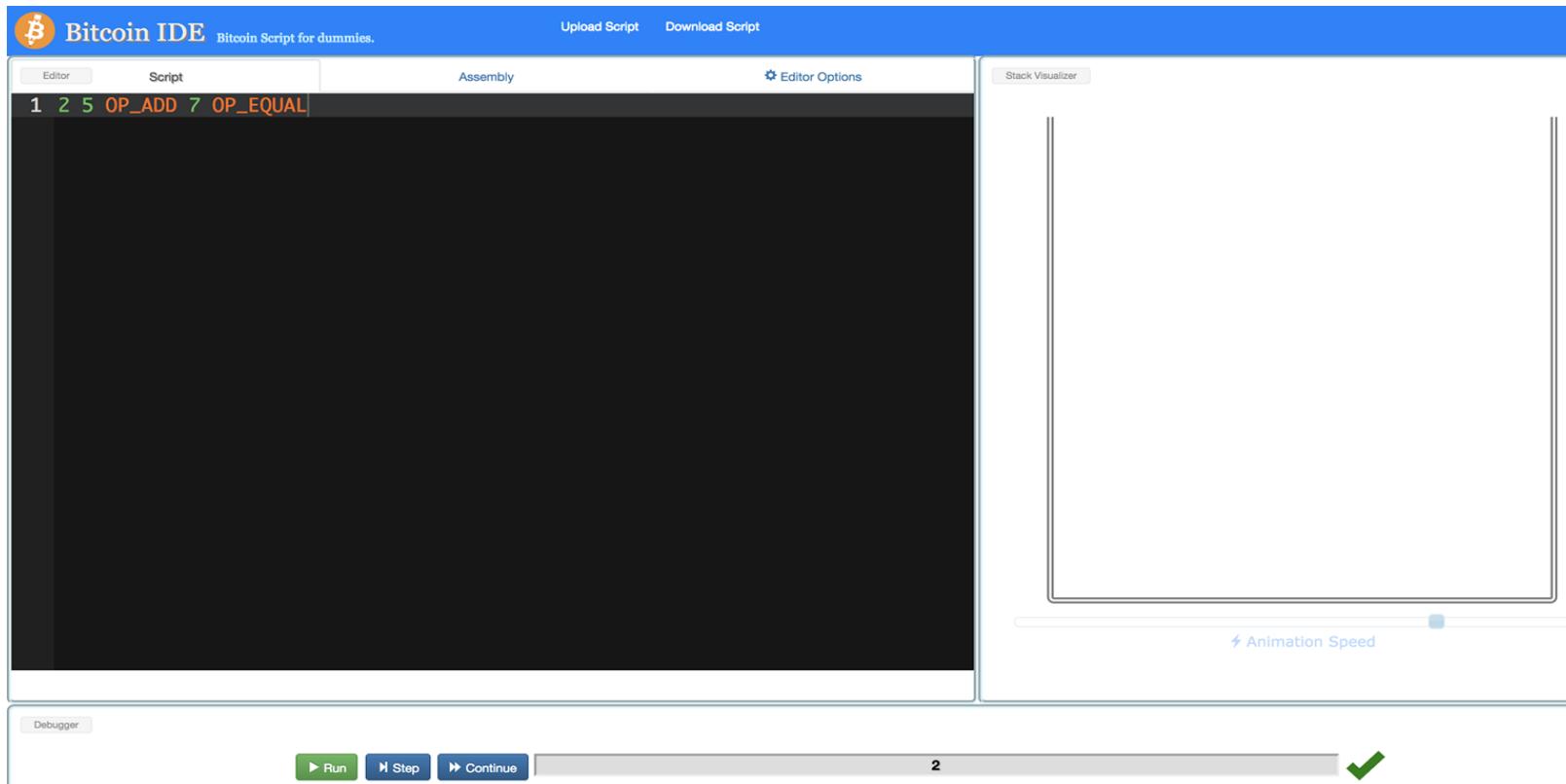
# Cómo funciona Bitcoin: transacciones



# Cómo funciona Bitcoin: recompensa a los mineros



# Cómo funciona Bitcoin: bitcoin scripting



# Cómo funciona Bitcoin: bitcoin scripting

## P2PKH (pay-to-public-key-hash)

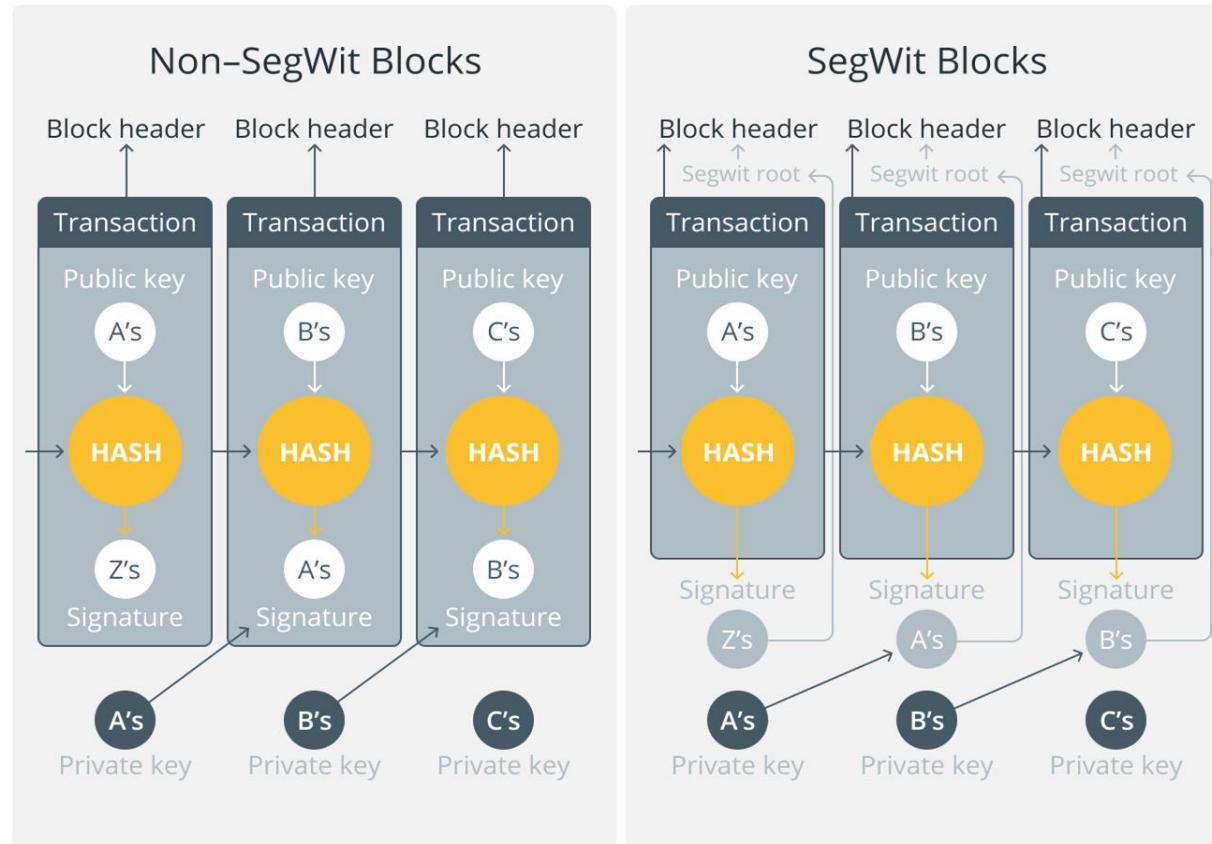
- ▶ *Script de bloqueo (scriptPubKey) = condición para usar el output*  
OP\_DUP OP\_HASH160 <pubKeyHash> OP\_EQUALVERIFY OP\_CHECKSIG
- ▶ *Script de desbloqueo (scriptSig) = prueba para usar el input*  
<sig> <pubkey>

Standard Script: P2PKH



[learnmeabitcoin.com](http://learnmeabitcoin.com)

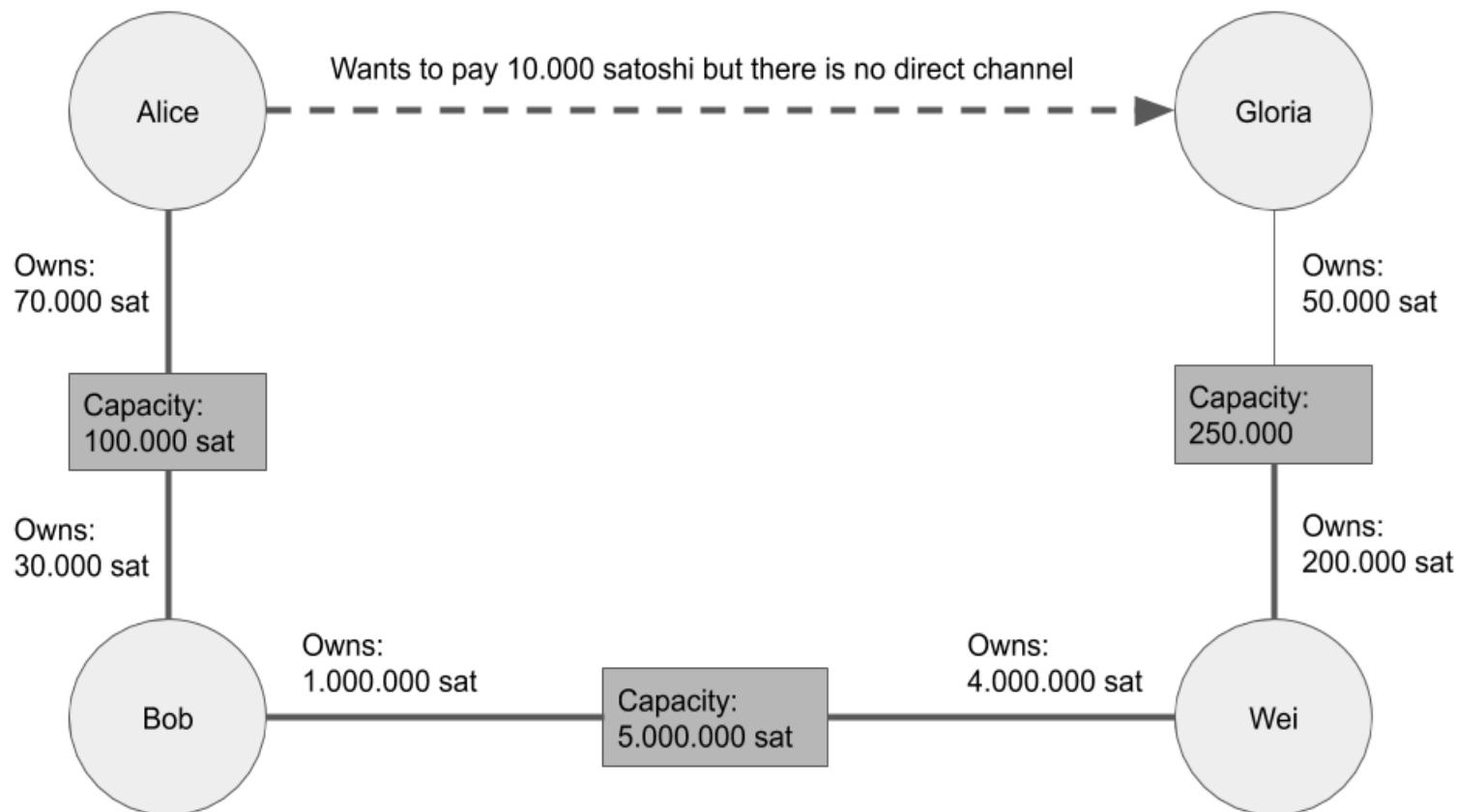
# Mejoras para incrementar el rendimiento: SegWit



 | cointelegraph.com

source: *Cointelegraph Analytics*

# Mejoras para incrementar el rendimiento: LN





# **TALLER PRÁCTICO: DESPLEGANDO UNA RED DE BLOCKCHAIN**

# Taller práctico

Para comprender el funcionamiento de una red pública de Blockchain vamos a descargar e instalar el software de bitcoin, creando una red privada y simulando tanto los procesos de minería como la generación de distintas operaciones que nos permitirán entender cómo funciona la tecnología que hace posibles las criptomonedas.

Para realizar esta práctica necesitaremos un entorno con Linux (también es posible en la consola de [Ubuntu sobre Windows10](#)) y acceso a Internet, donde instalaremos [bitcoind](#) en modo [regtest](#), lo que nos permitirá construir una red privada con varios nodos.

# Taller práctico: descarga del software

1. Descargar el software de bitcoin desde <https://bitcoin.org/en/download> teniendo en cuenta la versión que se requiere en función del entorno (Linux si utilizas el subsistema Ubuntu sobre Windows, ARM Linux 64 bits si utilizas RaspberryPi con Ubuntu).



En Ubuntu sobre Windows:

```
jorge@jorge-rpi:~$ wget https://bitcoin.org/bin/bitcoin-core-22.0/bitcoin-22.0-x86_64-linux-gnu.tar.gz
```

# Taller práctico: descomprimir el software

2. Descomprimiremos en nuestro \$HOME:

```
jorge@jorge-rpi:~$ tar xvf bitcoin*.tar.gz -C $HOME
```

3. Crearemos un enlace simbólico “bitcoin” en el \$HOME apuntando al directorio creado al descomprimir:

```
jorge@jorge-rpi:~$ ln -s $HOME/bitcoin-22.0 $HOME/bitcoin
```

4. Añadiremos al PATH el directorio donde se encuentran los binarios de bitcoin

```
jorge@jorge-rpi:~$ export PATH=$PATH:$HOME/bitcoin/bin
```

# Taller práctico: ejecutar dos nodos de bitcoin

5. Crearemos dos directorios de datos diferentes para poder ejecutar simultáneamente dos instancias de bitcoind en nuestro entorno y simular una red de nodos:

```
jorge@jorge-rpi:~$ mkdir $HOME/bitcoin/nodo1  
  
jorge@jorge-rpi:~$ mkdir $HOME/bitcoin/nodo2
```

6. Lanzaremos los dos procesos de bitcoind, cada uno escuchando en un puerto distinto y utilizando un directorio de datos diferente:

```
jorge@jorge-rpi:~$ bitcoind -regtest -port=1234 -datadir=$HOME/bitcoin/nodo1 -rpcport=1234 -  
bind=127.0.0.1:1235=onion --daemon  
  
jorge@jorge-rpi:~$ bitcoind -regtest -port=2345 -datadir=$HOME/bitcoin/nodo2 -rpcport=2345 -  
bind=127.0.0.1:2346=onion --daemon
```

# Taller práctico: el directorio regtest

7. Lanzados los procesos verificaremos qué se crea en el directorio “regtest” de cada nodo:

```
jorge@jorge-rpi:~$ ls $HOME/bitcoin/nodo1/regtest
```

Comprobaremos que se han creado varios ficheros, nos interesa en particular el fichero “debug.log” donde estarán las trazas del proceso bitcoind.

Mantendremos abierta una consola con un “tail -f” de ese fichero de cada nodo a lo largo de la práctica, para ver cómo se va actualizando la información a medida que el proceso bitcoind interacciona con otros nodos de la red, y responde a nuestros comandos.

# Taller práctico: el directorio regtest

También se crean varios directorios:

- En “blocks” y “chainstate” se almacena la información de la cadena de bloques (se trata en la práctica de una base de datos LevelDB que permite almacenar parejas clave-valor). Estos directorios pueden ser copiados a otras instancias para facilitar el proceso de sincronización (interesante en la mainnet).
- En “wallets” se guardan los monederos que creamos (que también pueden ser copiados, por ejemplo, para almacenar un backup en lugar seguro).

Para una descripción muy detallada de la información que se almacena en bitcoin: <https://bitcoindev.network/understanding-the-data/>

# Taller práctico: interactuando con bitcoin-cli

8. Crearemos dos alias para utilizar bitcoin-cli sin tener que repetir todos los parámetros (bitcoin-cli es una utilidad que nos permite interactuar desde linea de comandos con los nodos en ejecución mediante llamadas RPC a los procesos bitcoind):

```
jorge@jorge-rpi:~$ nodo1="bitcoin-cli -regtest -datadir=$HOME/bitcoin/nodo1 -rpcport=1234"  
jorge@jorge-rpi:~$ nodo2="bitcoin-cli -regtest -datadir=$HOME/bitcoin/nodo2 -rpcport=2345"
```

# Taller práctico: interactuando con bitcoin-cli

9. Comprobaremos con una llamada RPC a cada uno de los procesos bitcoind que se están ejecutando que todo está OK:

```
jorge@jorge-rpi:~$ $nodo1 getblockchaininfo  
jorge@jorge-rpi:~$ $nodo2 getblockchaininfo
```

OJO: ¡No olvidar el \$ inicial para ejecutar el comando!

Comprobaremos en la salida cómo la cadena en ambos casos es “regtest” y no se han generado todavía bloques.

El listado de métodos que podemos utilizar en bitcoin-cli está en  
[https://en.bitcoin.it/wiki/Original\\_Bitcoin\\_client/API\\_calls\\_list](https://en.bitcoin.it/wiki/Original_Bitcoin_client/API_calls_list))

# Taller práctico: crear un wallet

10. Crearemos un wallet en cada nodo, que llamaremos “wallet1” y “wallet2” respectivamente:

```
jorge@jorge-rpi:~$ $nodo1 createwallet "wallet1"  
  
jorge@jorge-rpi:~$ $nodo2 createwallet "wallet2"
```

Comprobaremos cómo en el directorio “wallets” de cada nodo se ha creado un nuevo directorio con el nombre del wallet y dentro de éste se habrá generado entre otros un archivo wallet.dat

Este fichero es el más relevante, ya que contiene las claves públicas y privadas, scripts correspondientes a la dirección de bitcoin (y otra información relacionada con el wallet).

# Taller práctico: conectando los nodos de la red

11. Registraremos nodo2 como peer del nodo1 (utilizaremos la IP de loopback para esta prueba, pero podríamos usar la IP que tengamos asignada en la red local si queremos conectar con nodos en otras máquinas):

```
jorge@jorge-rpi:~$ $nod01 addnode "127.0.0.1:2346" "add"
```

12. Comprobaremos que el nodo1 ve correctamente al nodo2:

```
jorge@jorge-rpi:~$ $nod01 getpeerinfo
```

# Taller práctico: crear el primer bloque (génesis)

13. Crearemos el primer bloque con el nodo1:

```
jorge@jorge-rpi:~$ $nodo1 -generate 1
```

En la salida del comando comprobaremos la dirección de bitcoin a la que se ha enviado el coinbase que se ha generado en el bloque, y el hash de dicho bloque, en mi caso:

```
{
  "address": "bcrt1q0gm0yeguhpa2jm7sy3t2ueh26qnuh872g777r8",
  "blocks": [
    "74bb1520f204ab70dd49d28691ba82403c791f85588fc809143b146c5067ed1e"
  ]
}
```

# Taller práctico: comprobar el balance del nodo1

14. Comprobaremos el balance del nodo1:

```
jorge@jorge-rpi:~$ $nodo1 getbalance
```

¿Por qué es 0, si hemos minado un bloque? Porque el coinbase sólo puede utilizarse una vez han pasado 100 confirmaciones... como medida de seguridad (mineros que puedan haber generado de forma maliciosa un bloque, por ejemplo para provocar un doble gasto, no pueden usar inmediatamente la recompensa).

Adicionalmente, en regtest (a diferencia de la mainnet) los primeros 150 bloques generados incluyen una recompensa de 50 bitcoins (a partir de ahí la recompensa se reduce a la mitad, mientras que en la mainnet esta reducción de la recompensa, llamado halving, sucede cada 4 años).

# Taller práctico: minar 100 bloques más

15. Minaremos otros 100 bloques en nodo1:

```
jorge@jorge-rpi:~$ $nodo1 -generate 100
```

16. Comprobaremos que el balance del nodo1 ya muestra los 50BTC correspondientes al primer bloque minado (que ya tiene 100 confirmaciones):

```
jorge@jorge-rpi:~$ $nodo1 getbalance
```

# Taller práctico: comprobar estado de la red

17. Comprobaremos cuál es el estado de la red en ambos nodos, nodo1 y nodo2:

```
jorge@jorge-rpi:~$ $nodo1 getblockchaininfo  
  
jorge@jorge-rpi:~$ $nodo2 getblockchaininfo
```

Deberían mostrar la misma información (observaremos en particular el número de bloques). Nota: el nodo2 puede tardar unos minutos en sincronizar.

# Taller práctico: comprobar balance del nodo2

18. Cargamos el wallet del nodo2 (el del nodo1 está ya cargado, al haber generado bloques). Este paso puede no ser necesario (si es así, recibiremos un aviso de que ya está cargado el wallet).

```
jorge@jorge-rpi:~$ $nodo2 loadwallet "wallet2"
```

19. Comprobamos el balance del nodo2 (debe ser 0):

```
jorge@jorge-rpi:~$ $nodo2 getbalance
```

# Taller práctico: enviar bitcoins al wallet2

20. Obtenemos una dirección del nodo2:

```
jorge@jorge-rpi:~$ $nodo2 getnewaddress "wallet2"
```

En mi caso obtengo: “bcrt1q8mln6xky7uh36g7ghh97tmztnu5xrgt5klc6h”

21. Enviamos 1 bitcoin del wallet del nodo1 (conseguido en el coinbase del primer bloque minado) a esta dirección del nodo2:

```
jorge@jorge-rpi:~$ $nodo1 -named sendtoaddress address="bcrt1q8mln6xky7uh36g7ghh97tmztnu5xrgt5klc6h"  
amount=1 fee_rate=25
```

Obtendremos como resultado el ID de la transacción (en mi caso, 9902155dd4151895095908857d54976989010e9b692fb14272322edb472320ce)

.

# Taller práctico: minar un nuevo bloque

22. Si comprobamos el balance del nodo2 veremos que NO se ha actualizado. ¿Por qué?

```
jorge@jorge-rpi:~$ $nodo2 getbalance
```

23. Porque necesita generarse un nuevo bloque que incluya esta transacción para que se actualice el balance (hasta ese momento la transacción no estará confirmada y no aparecerá por tanto en el balance del nodo2). Para ello minamos un nuevo bloque en el nodo1:

```
jorge@jorge-rpi:~$ $nodo1 -generate 1
```

# Taller práctico: comprobar el balance de los wallets

24. Ahora comprobamos nuevamente el balance de los dos wallets:

```
jorge@jorge-rpi:~$ $nodo1 getbalance
```

Obtendremos 48.99996475 (50 BTC - 1 BTC - la comisión que hemos indicado).

```
jorge@jorge-rpi:~$ $nodo2 getbalance
```

Obtendremos 1 BTC

# Taller práctico: parada de los nodos

25. Al finalizar, pararemos ambos procesos:

```
jorge@jorge-rpi:~$ $nodo1 stop  
  
jorge@jorge-rpi:~$ $nodo2 stop
```



Afi Escuela

---

© 2021 Afi Escuela. Todos los derechos reservados.