INTRODUCCIÓN A BLOCKCHAIN: CONOCIENDO LA TECNOLOGÍA

Open Innovation Campus

28 Abril 2022

Sobre mí…

Área de Blockchain
Telefónica

Alberto García García-Castro

@aggcastro

Bitcoin v0.1 released

Satoshi Nakamoto

“Announcing the first release of Bitcoin, a new electronic cash system that uses a peer-to-peer network to prevent double-spending. It's completely decentralized with no server or central authority.”

Telefónica @aggcastro

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 going through a financial institution. Digital signatures provide part of the solution, but the main benefits are lost if a trusted third party is still required to prevent double-spending.*

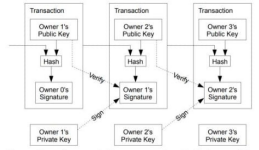
We propose a solution to the double-spending problem using a peer-to-peer network. The network timestamps transactions by hashing them into an ongoing chain of hash-based proof-of-work, forming a record that cannot be changed without redoing the proof-of-work. The longest chain not only serves as proof of the sequence of events witnessed, but proof that it came from the largest pool of CPU power. As long as a majority of CPU power is controlled by nodes that are not cooperating to attack the network, they'll generate the longest chain and outpace attackers. The network itself requires minimal structure. Messages are broadcast 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.

1. Introduction

Commerce on the Internet has come to rely almost exclusively on financial institutions serving as trusted third parties to process electronic payments. While the system works well enough for most transactions, it still suffers from the inherent weaknesses of the trust based model. Completely non-reversible transactions are not really possible, since financial institutions cannot avoid mediating disputes. The cost of mediation increases transaction costs, limiting the minimum practical transaction size and cutting off the possibility for small casual transactions, and there is a broader cost in the loss of ability to make non-reversible payments for nonreversible services. With the possibility of reversal, the need for trust spreads. Merchants must be wary of their customers, hassling them for more information than they would otherwise need. A certain percentage of fraud is accepted as unavoidable. These costs and payment uncertainties can be avoided in person by using physical currency, but no mechanism exists to make payments over a communications channel without a trusted party. What is needed is an electronic payment system based on cryptographic proof instead of trust, allowing any two willing parties to transact directly with each other without the need for a trusted third party. Transactions that are computationally impractical to reverse would protect sellers from fraud, and routine escrow mechanisms could easily be implemented to protect buyers. In this paper, we propose a solution to the double-spending problem using a peer-to-peer distributed timestamp server to generate computational proof of the chronological order of transactions. The system is secure as long as honest nodes collectively control more CPU power than any cooperating group of attacker nodes.

2. Transactions

We define an electronic coin as a chain of digital signatures. Each owner transfers the coin to the next by digitally signing a hash of the previous transaction and the public key of the next owner and adding these to the end of the coin. A payee can verify the signatures to verify the chain of ownership.



The problem of course is the payee can't verify that one of the owners did not double-spend the coin. A common solution is to introduce a trusted central authority, or mint, that checks every transaction for double spending. After each transaction, the coin must be returned to the mint to issue a new coin, and only coins issued directly from the mint are trusted not to be double-spent. The problem with this solution is that the fate of the entire money system depends on the company running the mint, with every transaction having to go through them, just like a bank. We need a way for the payee to know that the previous owners did not sign any earlier transactions. For our purposes, the earliest transaction is the one that counts, so we don't care about later attempts to double-spend. The only way to confirm the absence of a transaction is to be aware of all transactions. In the mint based model, the mint was aware of all transactions and decided which arrived first. To accomplish this without a trusted party, transactions must be publicly announced [1], and we need a system for participants to agree on a single history of the order in which they were received. The payee needs proof that at the time of each transaction, the

The proof-of-work also solves the problem of determining representation in majority decision making. If the majority were based on one-IP-address-one-vote, it could be subverted by anyone able to allocate many IPs. Proof-of-work is essentially one-CPU-one-vote. The majority decision is represented by the longest chain, which has the greatest proof-of-work effort invested in it. If a majority of CPU power is controlled by honest nodes, the honest chain will grow the fastest and outpace any competing chains. To modify a past block, an attacker would have to redo the proof-of-work of the block and all blocks after it and then catch up with and surpass the work of the honest nodes. We will show later that the probability of a slower attacker catching up diminishes exponentially as subsequent blocks are added. To compensate for increasing hardware speed and varying interest in running nodes over time, the proof-of-work difficulty is determined by a moving average targeting an average number of blocks per hour. If they're generated too fast, the difficulty increases.

5. Network

The steps to run the network are as follows:

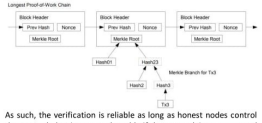
1) New transactions are broadcast to all nodes.
2) Each node collects new transactions into a block.
3) Each node works on finding a difficult proof-of-work for its block.
4) When a node finds a proof-of-work, it broadcasts the block to all nodes.
5) Nodes accept the block only if all transactions in it are valid and not already spent.
6) Nodes express their acceptance of the block by working on creating the next block in the chain, using the hash of the accepted block as the previous hash.

Nodes always consider the longest chain to be the correct one and will keep working on extending it. If two nodes broadcast different versions of the next block simultaneously, some nodes may receive one or the other first. In that case, they work on the first one they received, but save the other branch in case it becomes longer. The tie will be broken when the next proof-of-work is found and one branch becomes longer; the nodes that were working on the other branch will then switch to the longer one. New transaction broadcasts do not necessarily need to reach all nodes. As long as they reach many nodes, they will get into a block before long. Block broadcasts are also tolerant of dropped messages. If a node does not receive a block, it will request it when it receives the next block and realizes it missed one.

6. Incentive

By convention, the first transaction in a block is a special transaction that starts a new coin owned by the creator of the block. This adds an incentive for nodes to support the network, and provides a way to initially distribute coins into circulation, since there is no central authority to issue them. The steady addition of a constant of amount of new coins is analogous to gold miners expending resources to add gold to circulation. In our case, it is CPU time and electricity that is expended. The incentive can also be funded with transaction fees. If the output value of a transaction is less than its input value, the difference is a transaction fee that is added to the incentive value of the block containing the transaction. Once a predetermined number of coins have entered circulation, the incentive can transition entirely to transaction fees and be completely inflation free. The incentive may help encourage nodes to stay honest. If a greedy attacker is able to assemble more CPU power than all the honest nodes, he would have to choose between using it to defraud people by stealing back his payments, or using it to generate new coins. He ought to find it more profitable to play by the rules, such rules that favour him with more new coins than everyone else combined, than to undermine the system and the

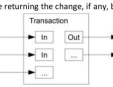
Merkle branch linking the transaction to the block it's timestamped in. He can't check the transaction for himself, but by linking it to a place in the chain, he can see that a network node has accepted it, and blocks added after it further confirm the network has accepted it.



As such, the verification is reliable as long as honest nodes control the network, but is more vulnerable if the network is overpowered by an attacker. While network nodes can verify transactions for themselves, the simplified method can be fooled by an attacker's fabricated transactions for as long as the attacker can continue to overpower the network. One strategy to protect against this would be to accept alerts from network nodes when they detect an invalid block, prompting the user's software to download the full block and alerted transactions to confirm the inconsistency. Businesses that receive frequent payments will probably still want to run their own nodes for more independent security and quicker verification.

9. Combining and Splitting Value

Although it would be possible to handle coins individually, it would be unwieldy to make a separate transaction for every cent in a transfer. To allow value to be split and combined, transactions contain multiple inputs and outputs. Normally there will be either a single input from a larger previous transaction or multiple inputs combining smaller amounts, and at most two outputs: one for the payment, and one returning the change, if any, back to the sender.



It should be noted that fan-out, where a transaction depends on several transactions, and those transactions depend on many more, is not a problem here. There is never the need to extract a complete standalone copy of a transaction's history.

10. Privacy

The traditional banking model achieves a level of privacy by limiting access to information to the parties involved and the trusted third party. The necessity to announce all transactions publicly precludes this method, but privacy can still be maintained by breaking the flow of information in another place: by keeping public keys anonymous. The public can see that someone is sending an amount to someone else, but without information linking the transaction to anyone. This is similar to the level of information released by stock exchanges, where the time and size of individual trades, the "tape", is made public, but without telling who the parties were.

As an additional firewall, a new key pair should be used for each transaction to keep them from being linked to a common owner. Some linking is still unavoidable with multi-input transactions, which necessarily reveal that their inputs were owned by the same owner. The risk is that if the owner of a key is revealed, linking could reveal other transactions that belonged to the same owner.

the sender can't change the transaction. We assume the sender is an attacker who wants to make the recipient believe he paid him for a while, then switch it to pay back to himself after some time has passed. The receiver will be alerted when that happens, but the sender hopes it will be too late. The receiver generates a new key pair and gives the public key to the sender shortly before signing. This prevents the sender from preparing a chain of blocks ahead of time by working on it continuously until he is lucky enough to get far enough ahead, then executing the transaction at that moment. Once the transaction is sent, the dishonest sender starts working in secret on a parallel chain containing an alternate version of his transaction. The recipient waits until the transaction has been added to a block and z blocks have been linked after it. He doesn't know the exact amount of progress the attacker has made, but assuming the honest blocks took the average expected time per block, the attacker's potential progress will be a Poisson distribution with expected value:

$$\lambda = \frac{z q}{p}$$

To get the probability the attacker could still catch up now, we multiply the Poisson density for each amount of progress he could have made by the probability he could catch up from that point:

$$\sum_{k=0}^{\infty} \frac{\lambda^k e^{-\lambda}}{k!} \cdot \begin{cases} (q/p)^{(z-k)} & \text{if } k \le z \\ 1 & \text{if } k > z \end{cases}$$

Rearranging to avoid summing the infinite tail of the distribution...

$$1 - \sum_{k=0}^{z} \frac{\lambda^k e^{-\lambda}}{k!} \left(1 - (q/p)^{(z-k)}\right)$$

Converting to C code...

```
#include <math.h>
double AttackerSuccessProbability(double q, int z)
{
    double p = 1.0 - q;
    double lambda = z * (q / p);
    double sum = 1.0;
    int i, k;
    for (k = 0; k <= z; k++)
    {
        double poisson = exp(-lambda);
        for (i = 1; i <= k; i++)
            poisson *= lambda / i;
        sum -= poisson * (1 - pow(q / p, z - k));
    }
    return sum;
}
```

Running some results, we can see the probability drop off exponentially with z. q=0.1 z=0 P=1.0000000 z=1 P=0.2045873 z=2 P=0.0509779 z=3 P=0.0131722 z=4 P=0.0034552 z=5 P=0.0009137 z=6 P=0.0002428 z=7 P=0.0000647 z=8 P=0.0000173 z=9 P=0.0000046 z=10 P=0.0000012 q=0.3 z=0 P=1.0000000 z=5 P=0.1773523 z=10 P=0.0416605 z=15 P=0.0101008 z=20 P=0.0024804 z=25 P=0.0006132 z=30 P=0.0001522 z=35 P=0.0000379 z=40 P=0.0000095 z=45 P=0.0000024 z=50 P=0.0000006

Solving for P less than 0.1%... P < 0.001 q=0.10 z=5 q=0.15 z=8 q=0.20 z=11 q=0.25 z=15 q=0.30 z=24 q=0.35 z=41 q=0.40 z=89 q=0.45 z=340

12. Conclusion

We have proposed a system for electronic transactions without relying on trust. We started with the usual framework of coins made from digital signatures, which provides strong control of ownership, but is incomplete without a way to prevent double-spending. To solve this, we proposed a peer-to-peer network using proof-of-work to record a public history of transactions that quickly becomes computationally impractical for an attacker to change if honest nodes control a majority of CPU power. The network is robust in its unstructured simplicity. Nodes work all at once with little coordination. They do not need to be identified, since

THE TIMES

Saturday January 3 2009

“Chancellor on brink of second bailout of banks”

Telefónica @aggcastro

Primera compra

22/05/2010

Laszlo Hanyecz compra dos pizzas por 10.000 BTC / 30$

Abierto

@alexandre_godreau from unsplash

Sin fronteras

@jakeblucker from unsplash

Neutral

@eberhardgross from unsplash

Resistente a la censura

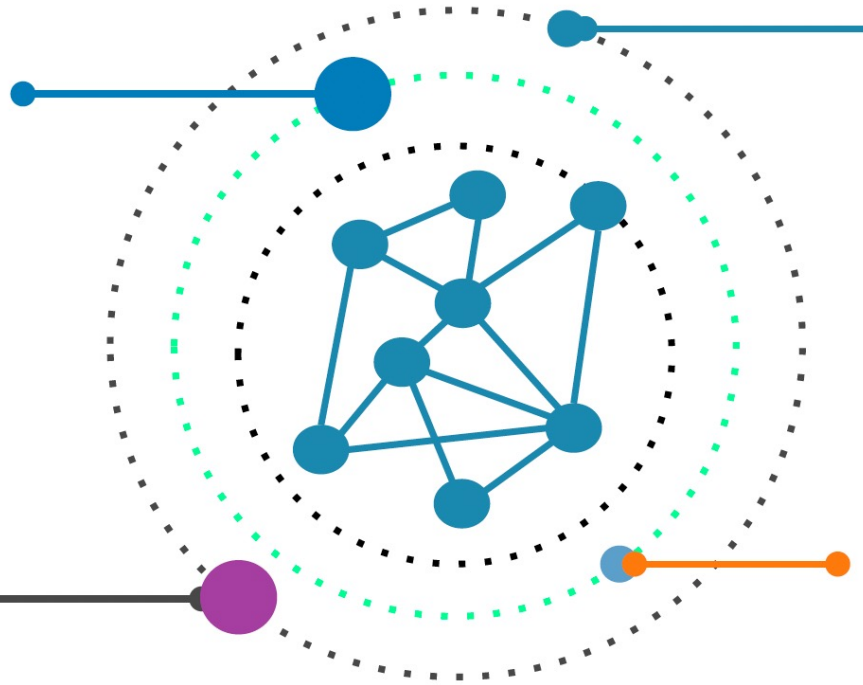
@markusspiske from unsplash

Público

@dulgier from unsplash

Al detalle

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



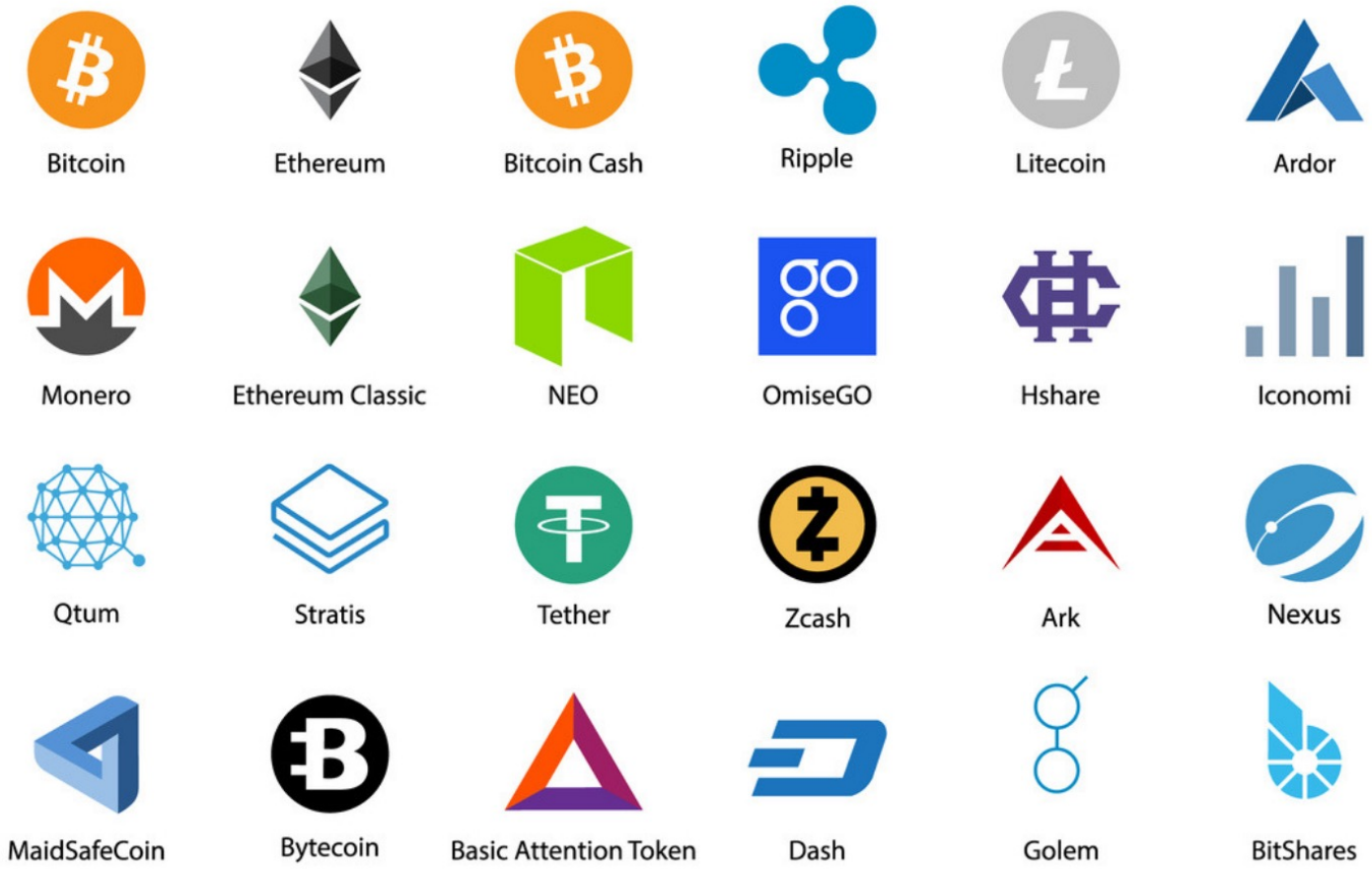
Los nodos de la red alcanzan un consenso sobre la información a registrar en cada momento

Hay diferentes tecnologías, públicas, privadas y permisionadas

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

Más Blockchain

En la actualidad existen miles de tipos de cadenas de bloques



Expectativas de negocio

IDC
ANALYZE THE FUTURE

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

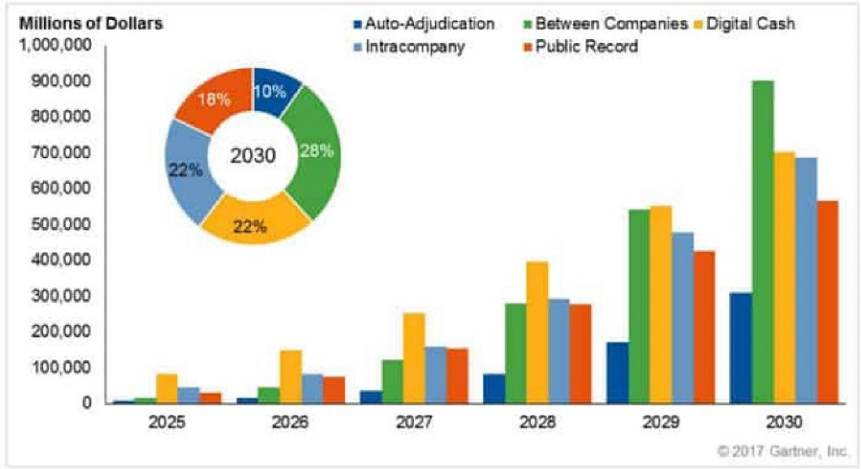
WORLD ECONOMIC FORUM

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

Gartner

"**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

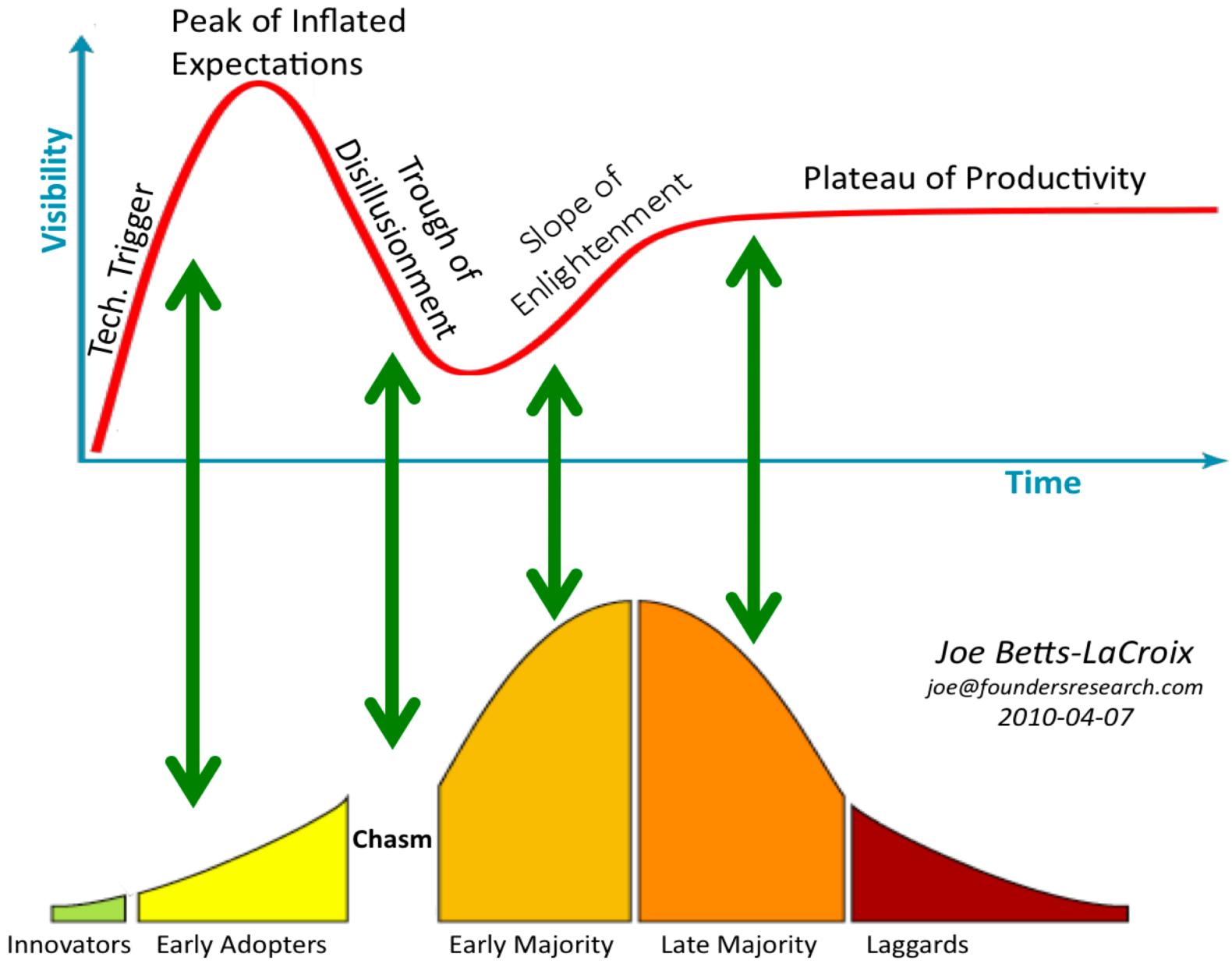


Source: Forecast: Blockchain Business Value, Worldwide, 2017-2030

Gartner

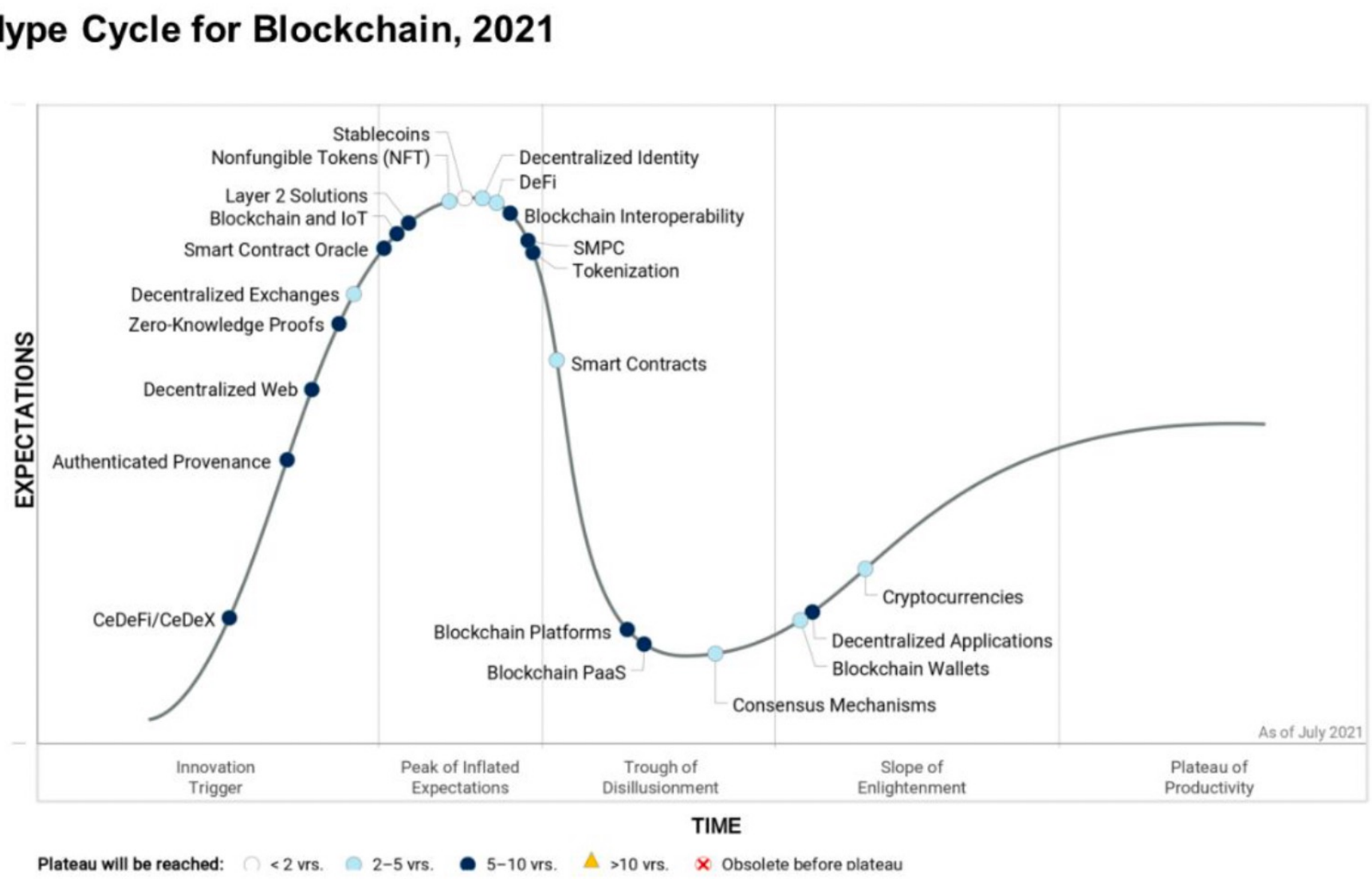
Hype Cycle

Representación gráfica de la madurez, adopción y aplicación comercial de una tecnología específica.



Gartner

Curva de tecnologías blockchain emergentes en 2021



Source: Gartner (July 2021)

747513

Blockchain pública vs privada

"Blockchain technology"

private (intra-)

Intranets & IT

Microsoft IBM ORACLE

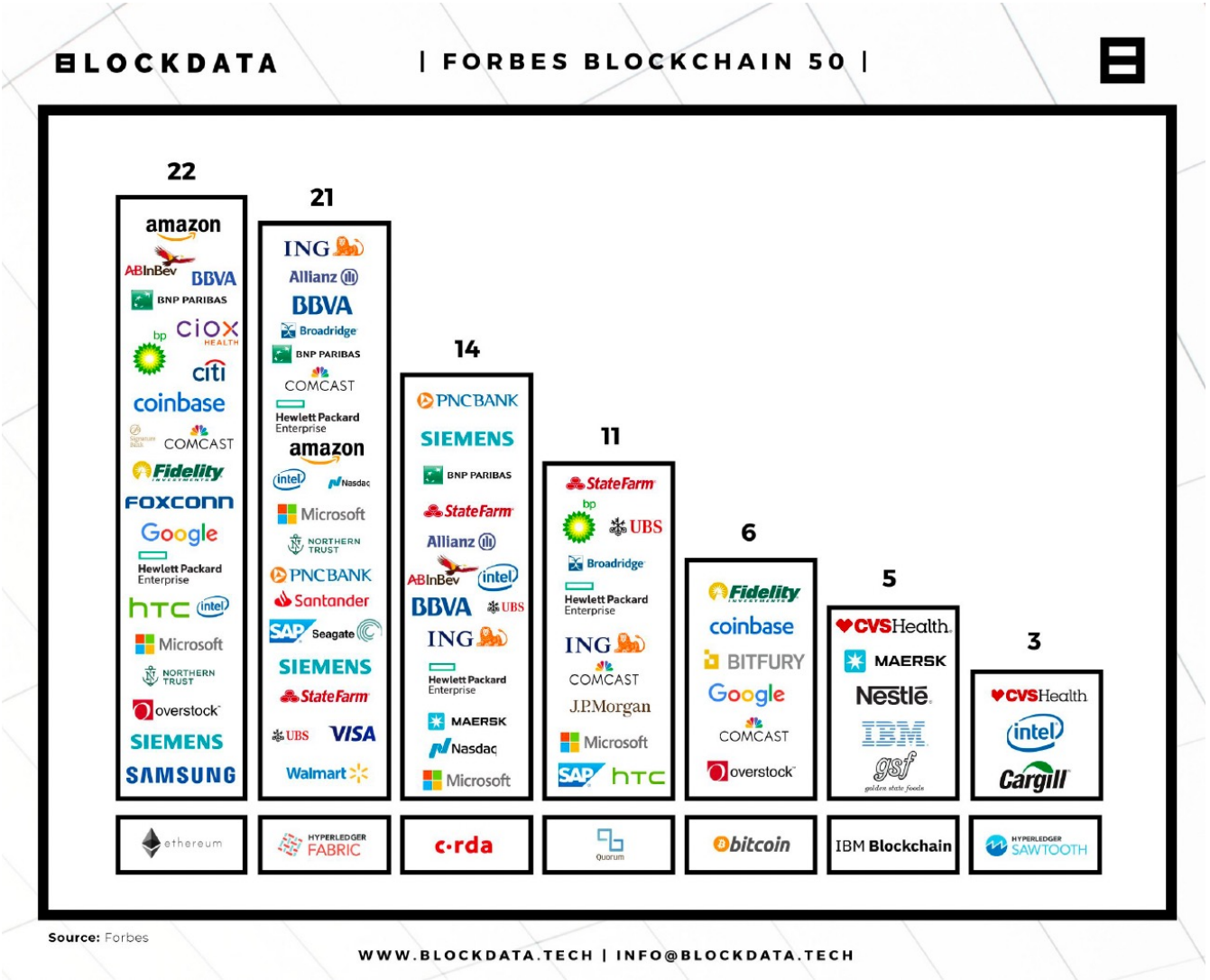
"The Bitcoin Blockchain"

public (inter-)

The Internet

facebook Google amazon

¿Hay más de una blockchain?

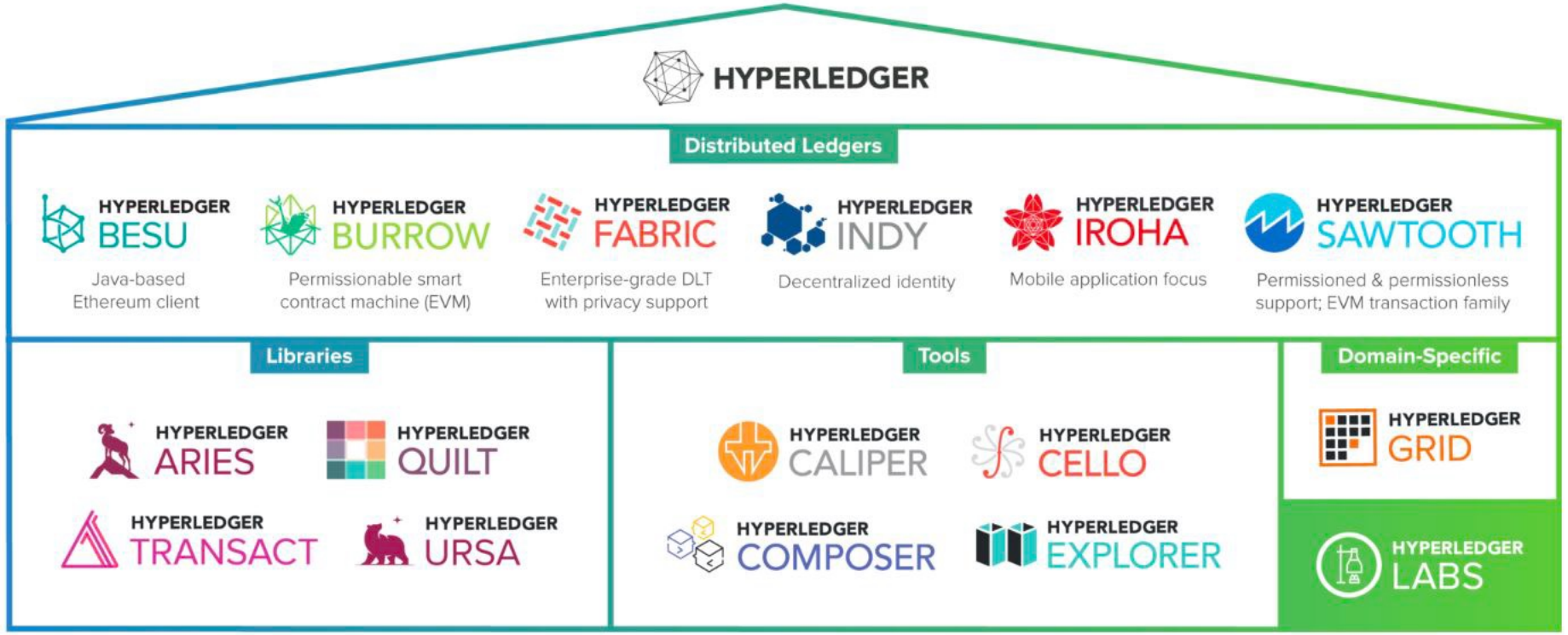


Telefónica @aggcastro

Ethereum

Plataforma global de código abierto para aplicaciones descentralizadas.

Hyperledger



Quorum:

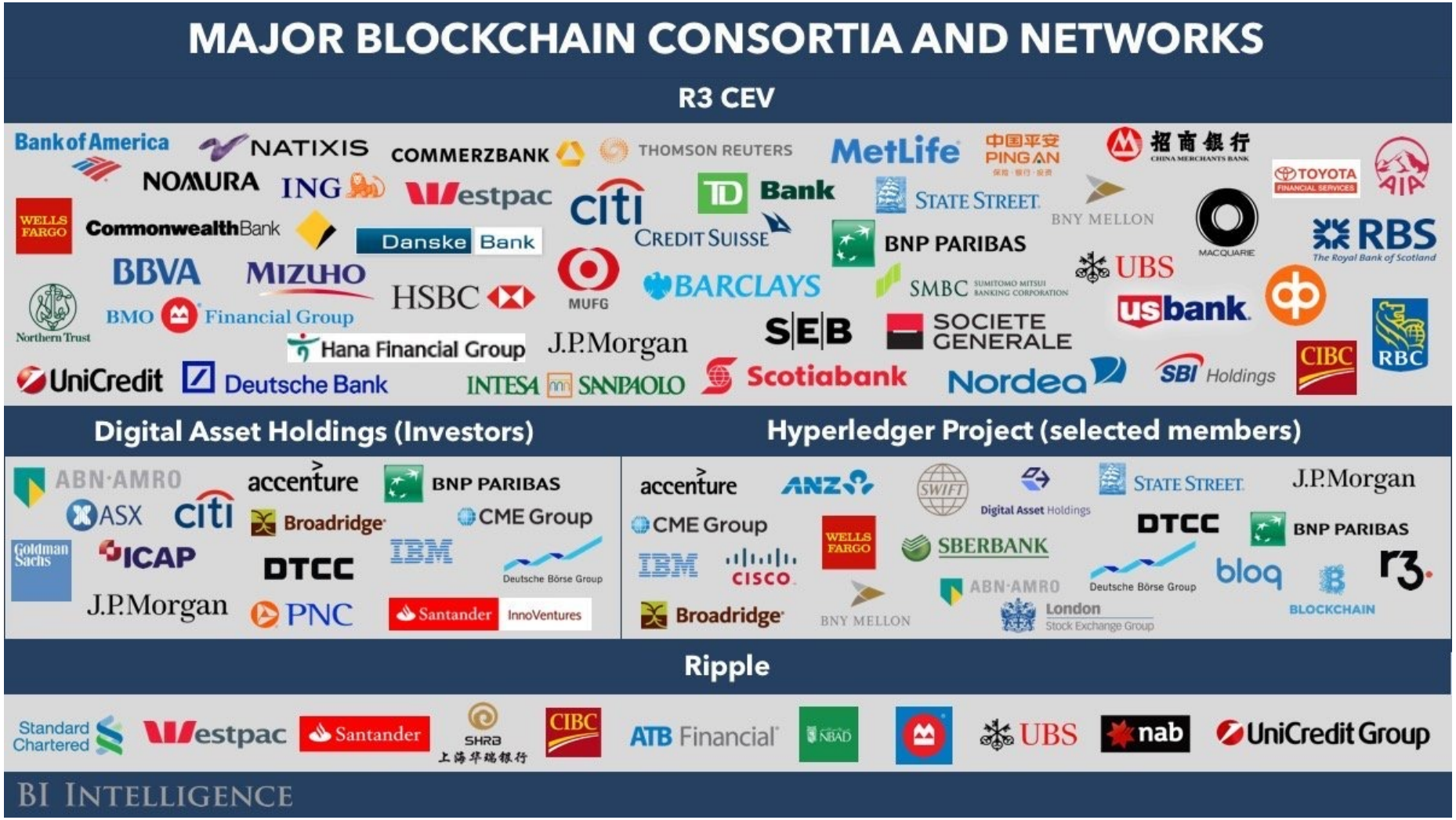
Ethereum for enterprise applications

jpmorgan.com/quorum

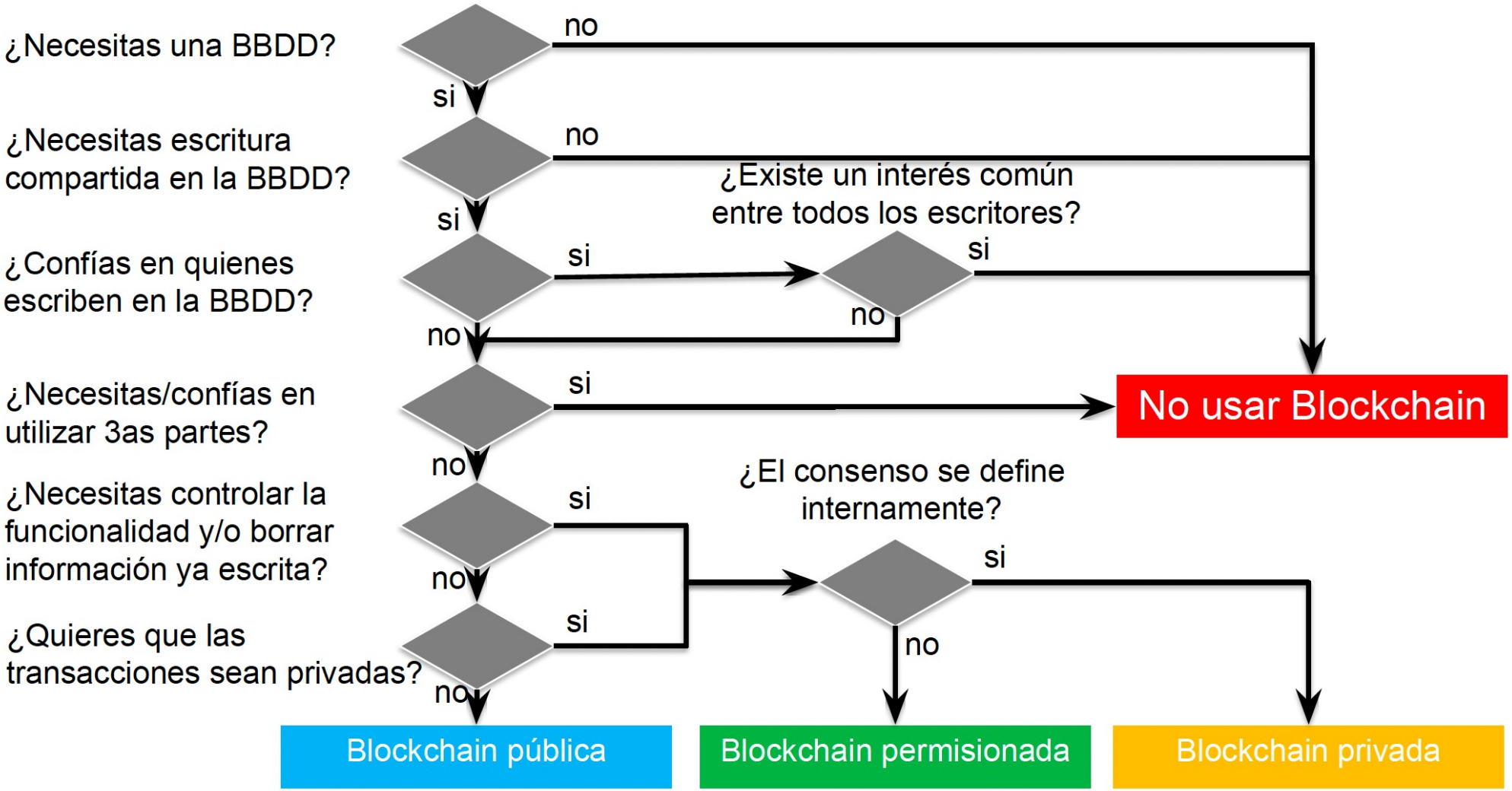
Consorcios

LACCHAIN

Consorcios

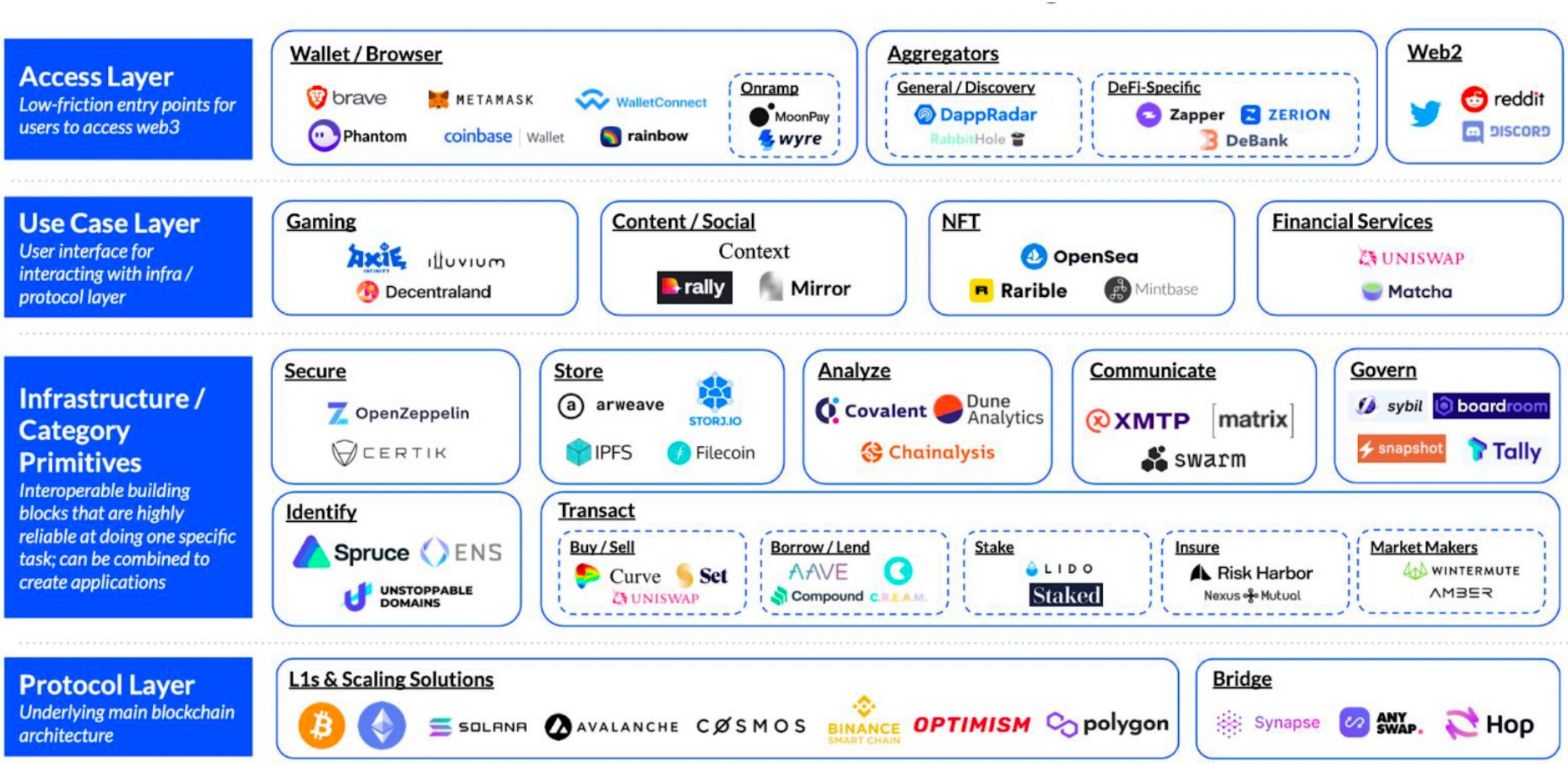

MAJOR BLOCKCHAIN CONSORTIA AND NETWORKS
R3 CEV
Bank of America
NATIXIS
COMMERZBANK
THOMSON REUTERS
MetLife
中国平安 PING AN
招商銀行 CHINA MERCHANTS BANK
TOYOTA FINANCIAL SERVICES
AIA
NOMURA
ING
Westpac
citi
TD Bank
STATE STREET
BNY MELLON
WELLS FARGO
CommonwealthBank
Danske Bank
CREDIT SUISSE
BNP PARIBAS
MACQUARIE
RBS The Royal Bank of Scotland
BBVA
MIZUHO
HSBC
MUFG
BARCLAYS
SMBC SUMITOMO MITSUI BANKING CORPORATION
UBS
Northern Trust
BMO Financial Group
usbank
Hana Financial Group
J.P.Morgan
SEB
SOCIETE GENERALE
CIBC
RBC
UniCredit
Deutsche Bank
INTESA SANPAOLO
Scotiabank
Nordea
SBI Holdings
Digital Asset Holdings (Investors)
ABN·AMRO
accenture
BNP PARIBAS
ASX
citi
Broadridge
CME Group
Goldman Sachs
ICAP
DTCC
IBM
Deutsche Börse Group
J.P.Morgan
PNC
Santander
InnoVentures
Hyperledger Project (selected members)
accenture
ANZ
SWIFT
Digital Asset Holdings
STATE STREET
J.P.Morgan
CME Group
DTCC
BNP PARIBAS
IBM
CISCO
WELLS FARGO
SBERBANK
ABN·AMRO
Deutsche Börse Group
bloq
r3
Broadridge
BNY MELLON
London Stock Exchange Group
BLOCKCHAIN
Ripple
Standard Chartered
Westpac
Santander
SHRB 上海华瑞银行
CIBC
ATB Financial
NBAD
UBS
nab
UniCredit Group
BI INTELLIGENCE

¿Usar blockchain?¿Cuál?



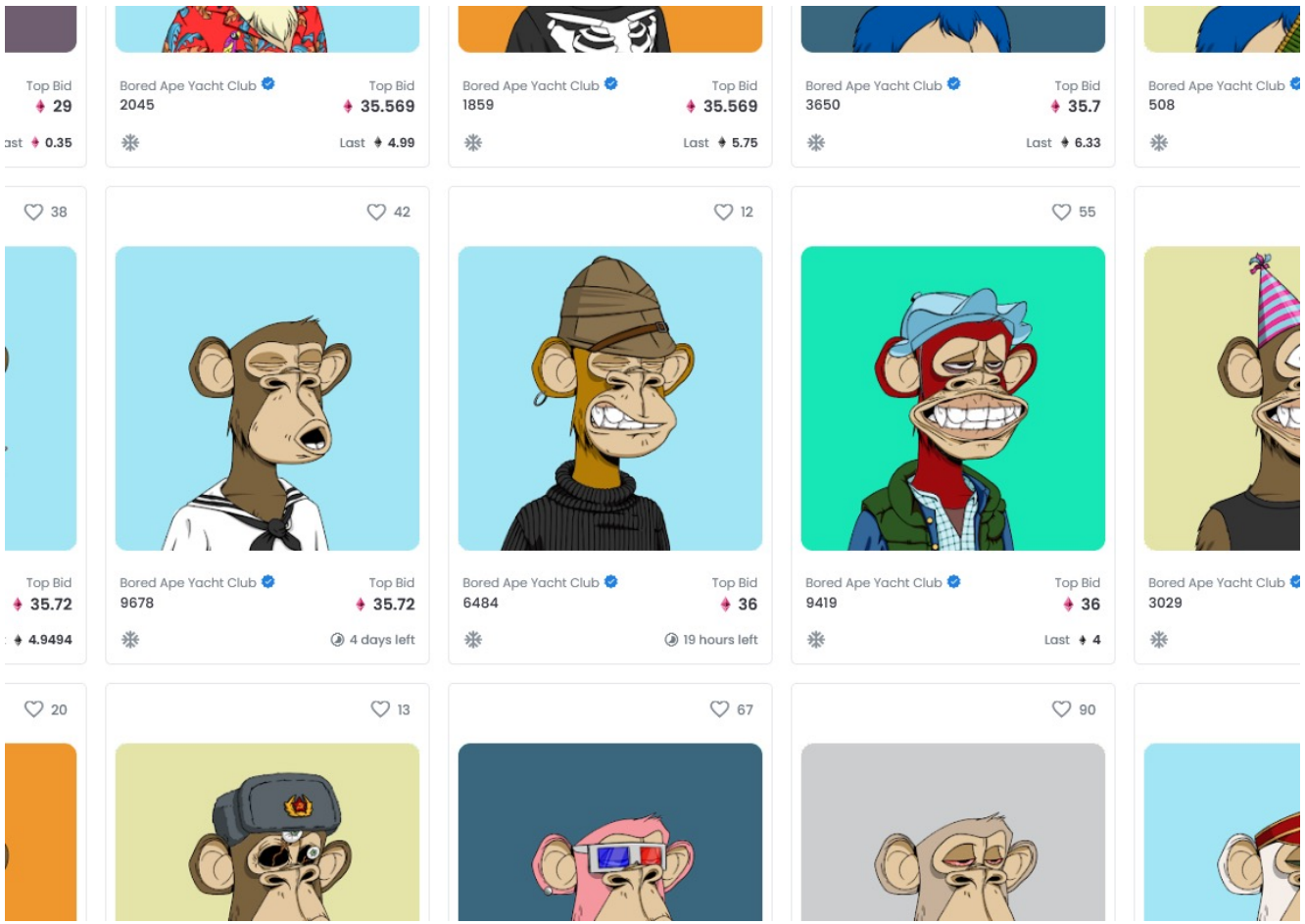
Cryptomonedas

DeFi



Telefónica @aggcastro

NFTs



Telefónica @aggcastro

Casos de uso

Casos de uso

¡Muchas gracias!