

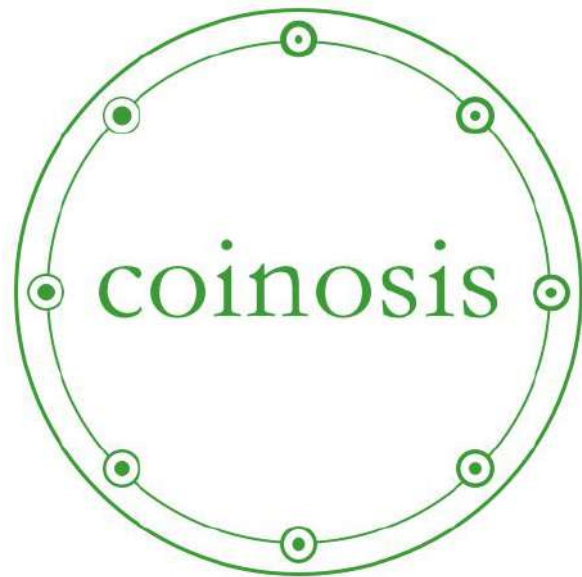
**TIC
TANK**

Blockchain de Ethereum como herramienta de gobernanza

26 de Agosto - 2020

Juan David Reyes

- Economista, Universidad de Los Andes
- Estudiante de Jurisprudencia, Universidad del Rosario
- Desarrollador front-end autodidacta
- Co-fundador de la plataforma educativa coinosis.co
- Co-organizador de Ethereum Bogotá



Motivación

- Retomaremos la primera charla de Tic Tank Talks: contratos inteligentes de Erick Rincón
- ¿Qué es la blockchain de Ethereum y cómo funciona?
- ¿Por qué ha sido útil para pensar soluciones descentralizadas?
- ¿Cómo esto abre nuevas implementaciones de gobernanza?
- Casos aplicados:
 - Mercados del arte
 - Transparencia en el manejo de recursos
 - Sistemas de incentivos en el sector educativo

Estructura de la charla

Pensando en la audiencia de TIC Tank, preparé la presentación entorno a 3 ideas:

- Desintermediación de mercados [Dada.art]
- Sistema de incentivos en educación [coinosis]
- Protocolos públicos de confianza [Programa de Alimentación Escolar]

Que tienen punto de encuentro para las personas que investigan y trabajan en Economía, Jurisprudencia, Matemáticas Aplicadas y Ciencias de la Computación.

¿Qué entenderemos por gobernanza?

En pocas palabras, la gobernanza se caracteriza por una red de instituciones e individuos que colaboran juntos y unidos por un pacto de mutua confianza, son organizaciones de poder que forman redes semiautónomas y a veces autogobernadas. La expresión de Rhodes (1996), "gobernanza sin gobierno", resume bien su concepción.

Gobernanza: una mirada desde América Latina (2011)

Cristina Zurbriggen

Índice

0. Conceptos importantes en Ethereum
 - a. Contratos inteligentes
 - b. Descentralización
 - c. Aplicaciones descentralizadas
1. Desintermediación de mercados
2. Diseño de incentivos en el sistema educativo
3. Corrupción en el manejo de recursos públicos
4. Desafíos, lecturas adicionales
5. Sorpresa para Colombia 🦄



Ideas del primer webinar


El contrato electrónico: la equivalencia funcional con la Ley 527 de 1999

1. Reconocimiento de la libertad de forma
2. Aplicación de las normas sustanciales
3. Validez de los contratos formados mediante mensajes de datos


“La mayoría de contratos nacen electrónicos.”

“Contract Legaltech Builders, todavía no hablamos de Smart Contracts.”

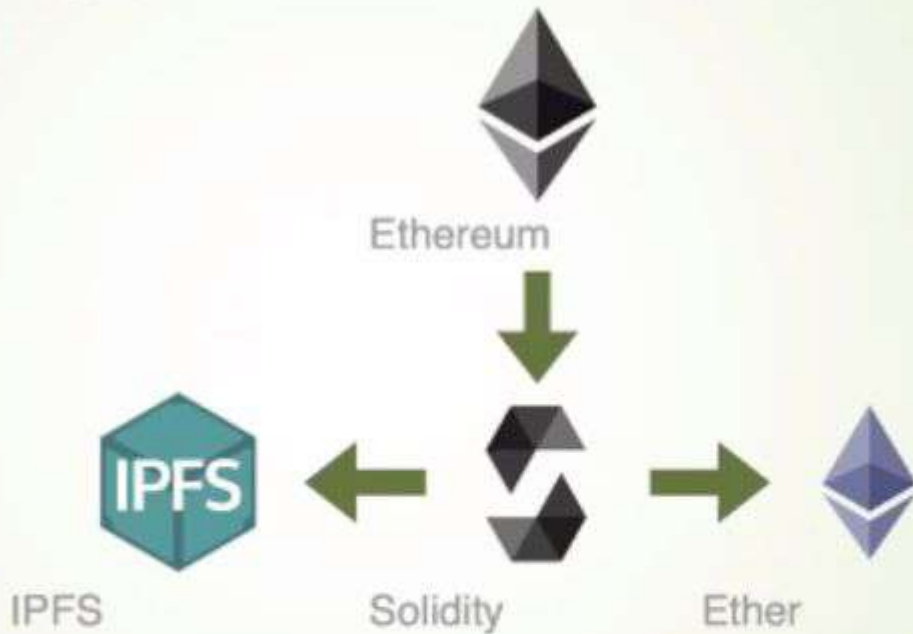
Principios de Blockchain

- 
- A vertical green line runs down the left side of the slide, featuring three white circles with green outlines and centers, spaced vertically.
- Integridad de la Red
 - Poder distribuido
 - El valor como incentivo
 - Seguridad
 - Privacidad
 - Derechos preservados
 - Inclusión

Índice

0. Conceptos importantes en Ethereum
 - a. Contratos inteligentes
 - b. Descentralización
 - c. Aplicaciones descentralizadas
1. Desintermediación de mercados
2. Diseño de incentivos en el sistema educativo
3. Corrupción en el manejo de recursos públicos
4. Desafíos, lecturas adicionales
5. Sorpresa para Colombia 

3. ¿Qué es Blockchain? – Ethereum



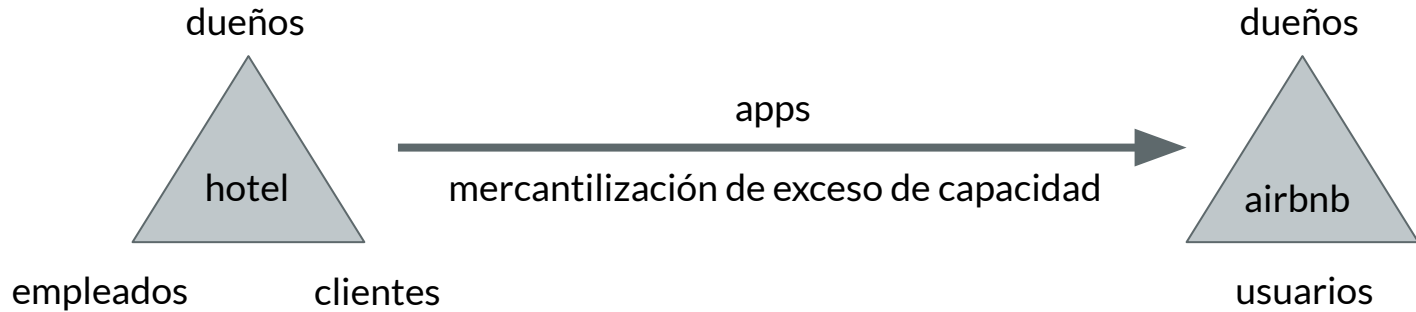
Qué es un Smart Contract



**Smart Contracts: Building Blocks
for Digital Markets (1996)**

Nick Szabo

la *sharing economy*

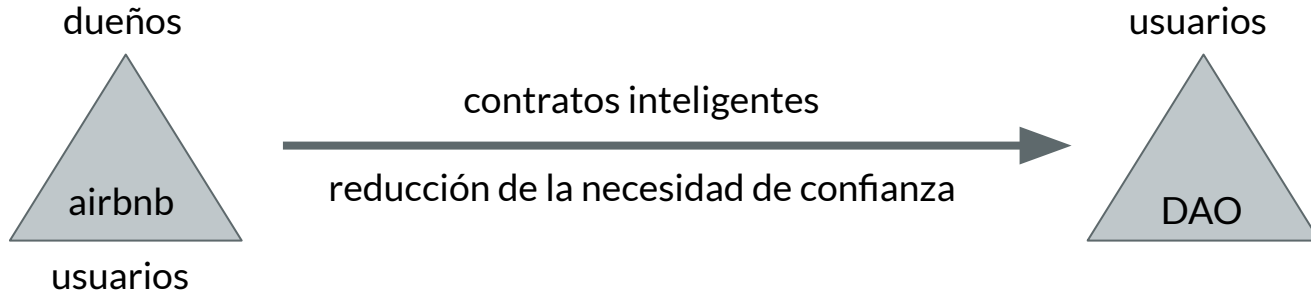




Costos de transacción

- triangulación: encontrar a la persona, establecer condiciones, fijar un precio
- transferencia: enviar el producto, enviar el dinero, devolver el producto
- confianza: qué hago si me estafa, devolución de dinero, compensación por daños

la *sharing economy* + blockchain



Cómo funciona la confianza

hotel

1. marca
2. hablar con el gerente
3. llamar a la policía
4. poner una demanda



airbnb

1. reviews
2. reportar usuario
3. resolución de conflictos
4. ¿llamar a la policía?
5. ¿poner una demanda?

Blockchain y la confianza

airbnb

1. reviews
2. reportar usuario
3. resolución de conflictos
4. ¿llamar a la policía?
5. ¿poner una demanda?



DAO

1. reviews
2. depósito de seguridad
3. cerradura inteligente
4. resolución comunitaria de conflictos

cómo funcionan las apps

front-end



back-end



cómo funcionan las dapps

front-end



back-end



servidor vs. blockchain

servidor

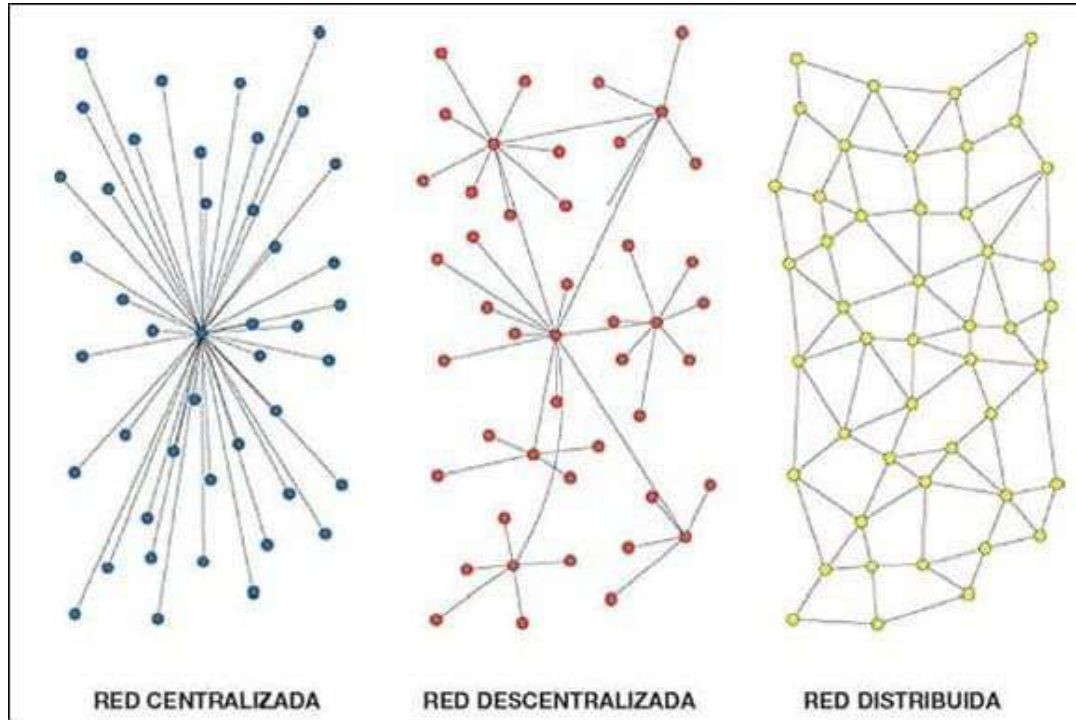
- le pertenece a una empresa
- está en un lugar
- unos pocos pueden ver los registros
- unos pocos pueden modificar los registros
- unos pocos pueden agregar registros
- usa contraseñas
- no maneja dinero



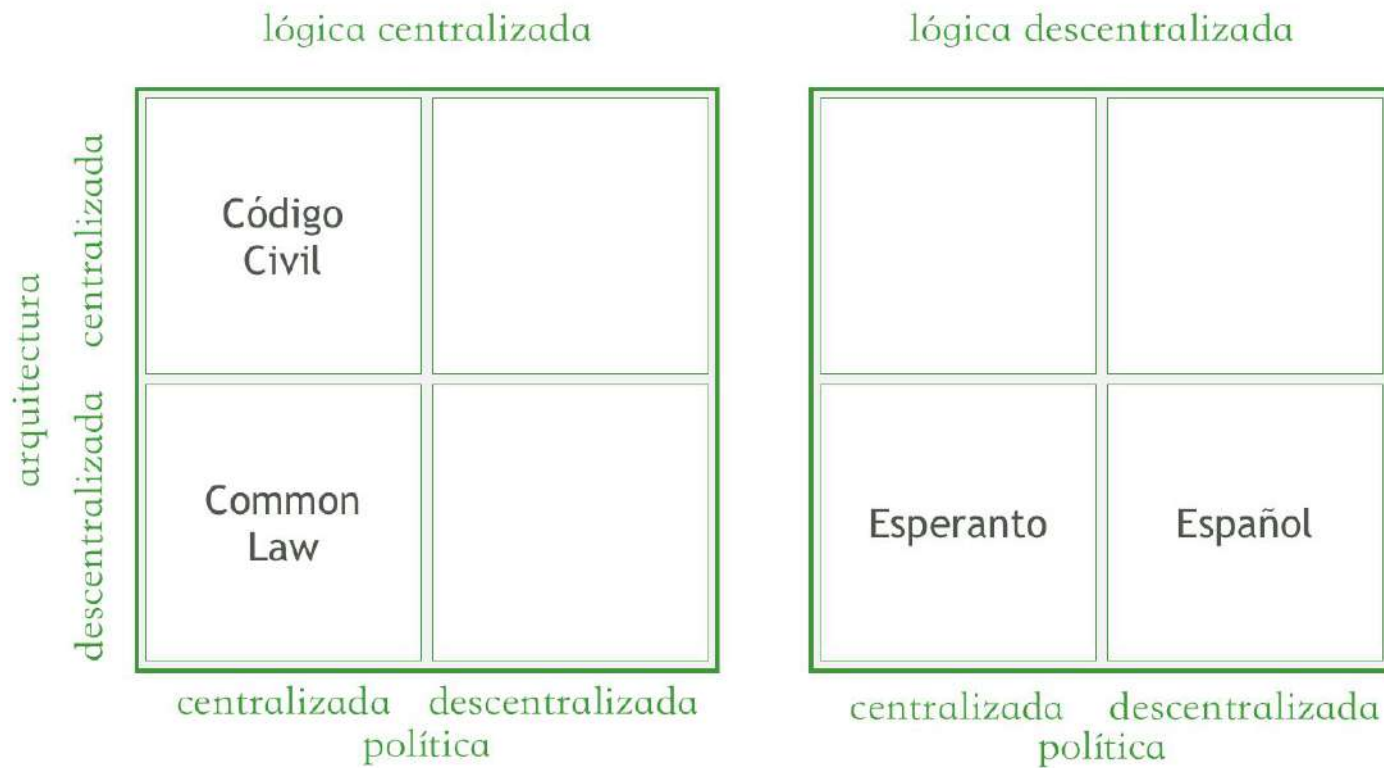
blockchain

- no le pertenece a nadie
- está en muchos lugares
- todos pueden ver los registros
- nadie puede modificar los registros
- algunos pueden agregar registros
- usa firmas digitales
- recibe, almacena y envía dinero

La visión básica de los sistemas distribuidos



La propuesta de Vitalik Buterin

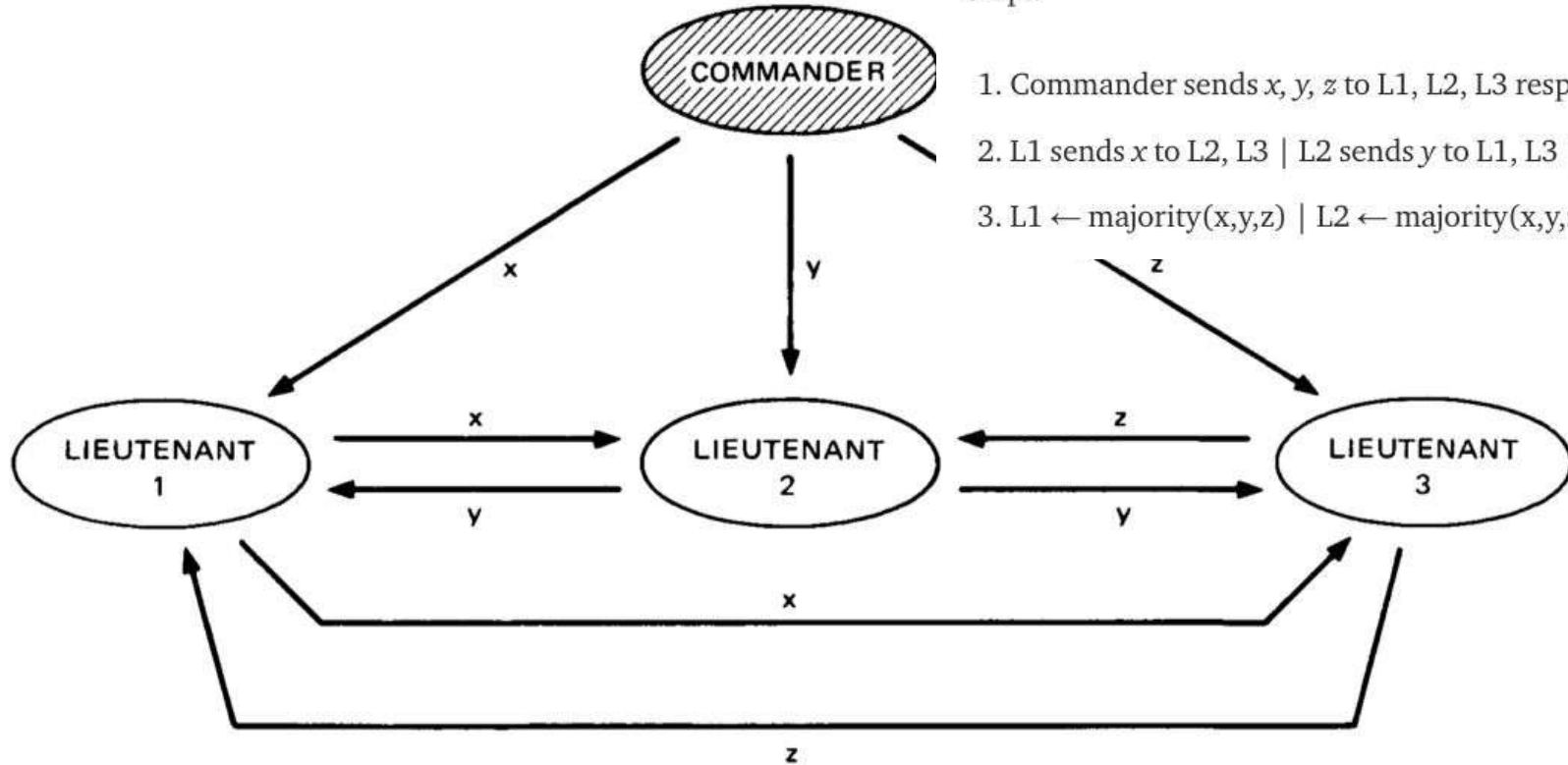




¿Cómo llegar al consenso en sistemas distribuidos?

“Un problema fundamental en los sistemas de computación distribuidos y además de múltiples agentes es lograr una construcción estable en presencia de nodos corruptos.”

El problema de los dos generales



Steps:

1. Commander sends x, y, z to L1, L2, L3 respectively

2. L1 sends x to L2, L3 | L2 sends y to L1, L3 | L3 sends z to L1, L2

3. L1 \leftarrow majority(x, y, z) | L2 \leftarrow majority(x, y, z) | L3 \leftarrow majority(x, y, z)

Introducción a la Teoría de Juegos

	Un Objetivo	Varios Objetivos
Un Decisor	Teoría de la Decisión Unipersonal	Teoría de la Decisión Multiobjetivo
Varios Decisores	Teoría de Juegos Cooperativos	Teoría de Juegos No Cooperativos

Dilema del prisionero

		PRISIONERO 2	
		Confesar	Mentir
PRISIONERO 1	Confesar	$(-8, -8)$	$(0, -10)$
	Mentir	$(-10, 0)$	$(-1, -1)$



Teoría de Juegos aplicada a la minería de Bitcoin

La minería es una competencia, los mineros gastan una gran cantidad de recursos, pues el PoW es una forma “probabilística” de tener un mecanismo Tolerante a la Falla Bizantina.

- Más billetes de lotería, mayores probabilidades de ganar.
- Más poder de cómputo, mayor probabilidad de minar el siguiente bloque.

Dinero programable: Ethereum como arena de juegos cooperativos

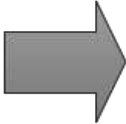
		B	
		COOP	DEFECT
A	COOP	1, 1	2, -3
	DEFECT	-3, 2	-1, -1

Original Prisoners' Dilemma Game

+

		B	
		COOP	DEFECT
A	COOP	0, 0	-1000, 0
	DEFECT	0, -1000	-1000, -1000

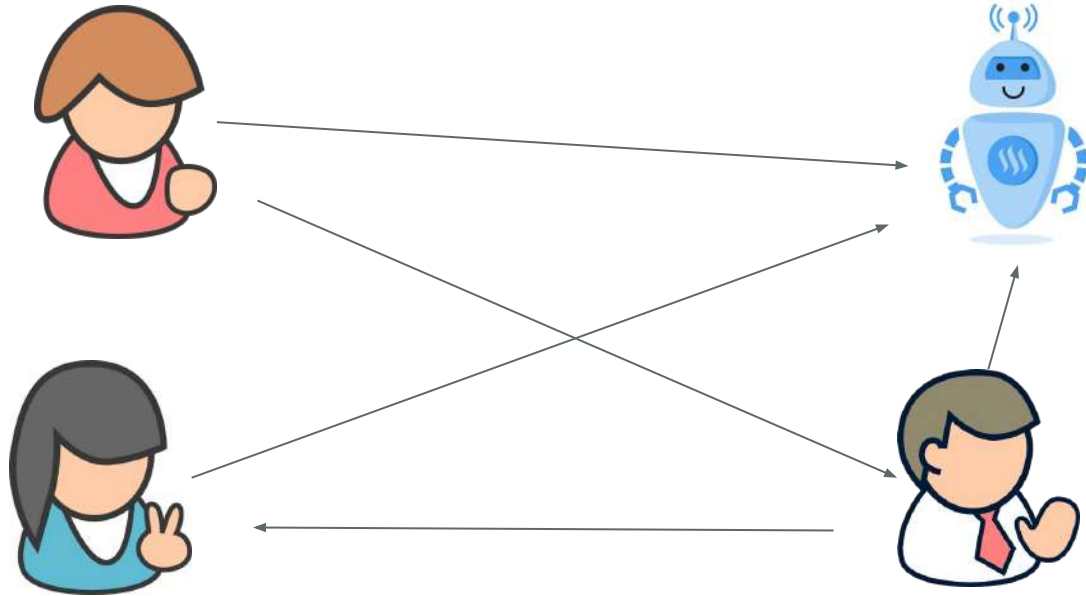
Ethereum contract of burning any player who chooses to defect



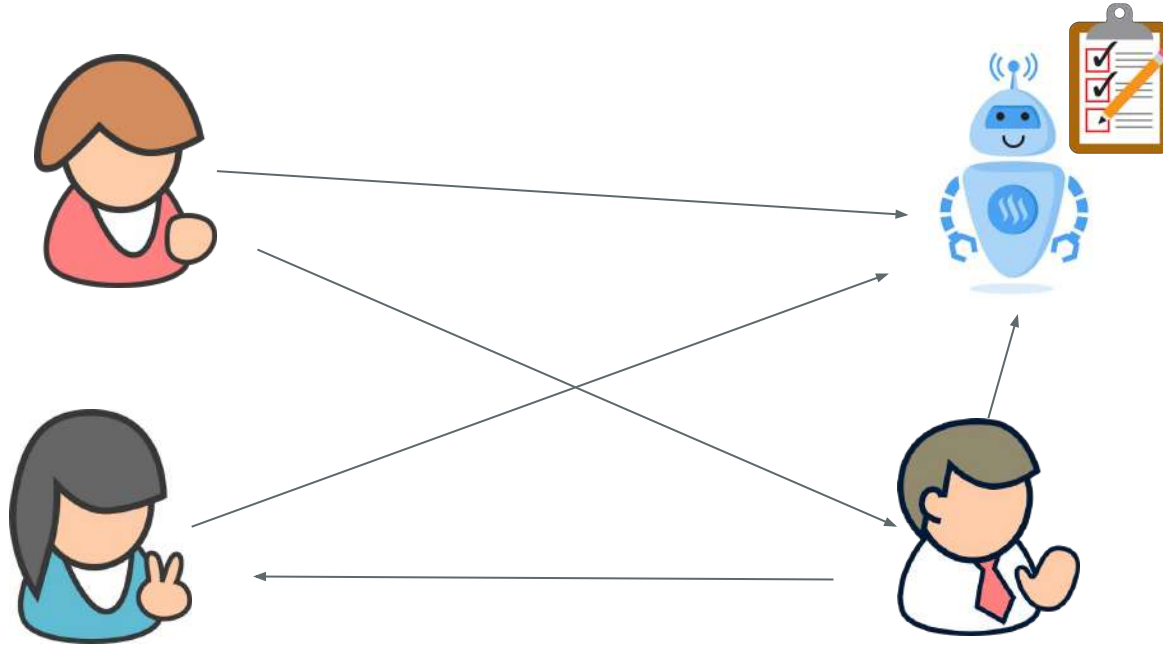
		B	
		COOP	DEFECT
A	COOP	1, 1	-998, -3
	DEFECT	-3, -998	-1000, -1001

Composite Game

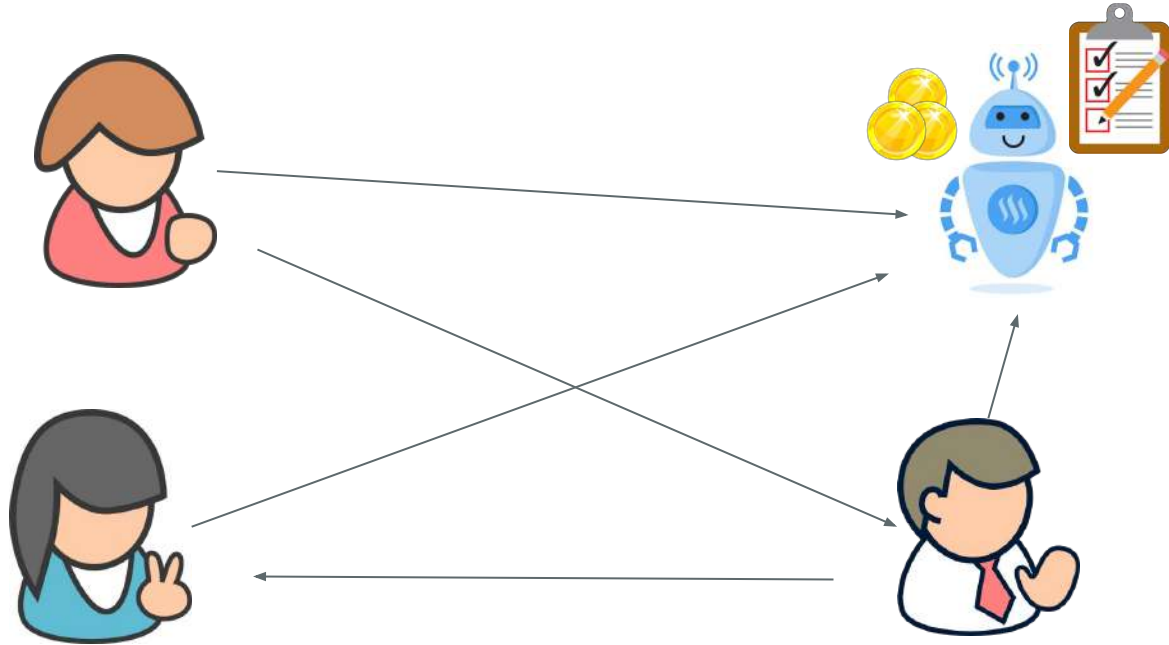
algunos usuarios no son humanos



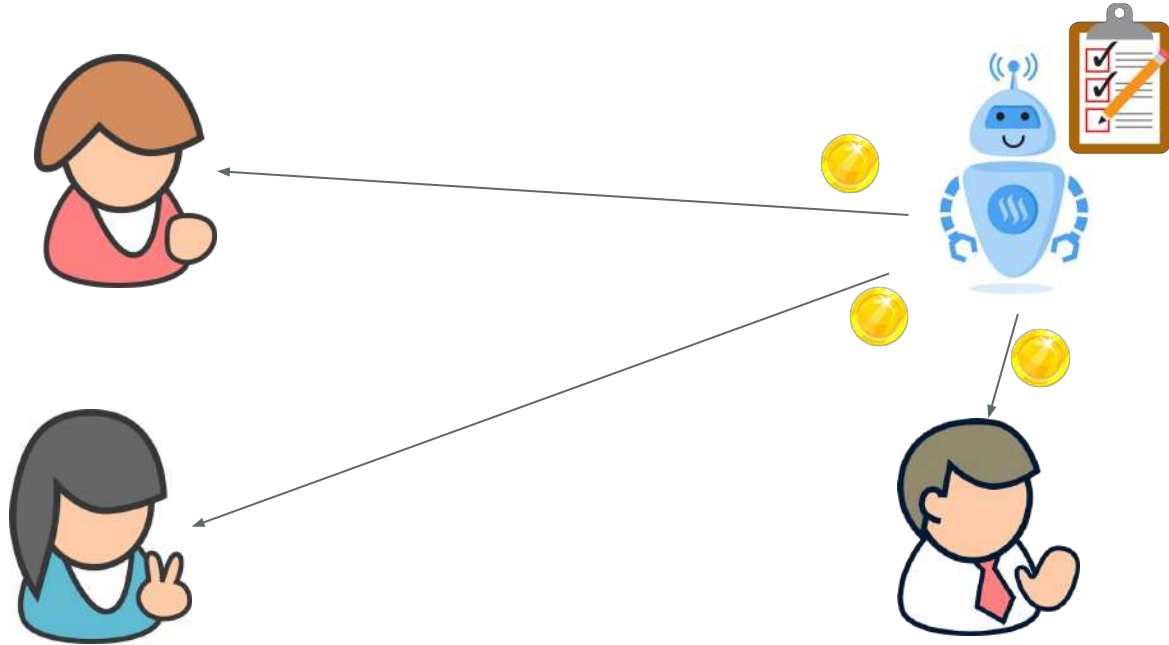
son programas que siguen reglas



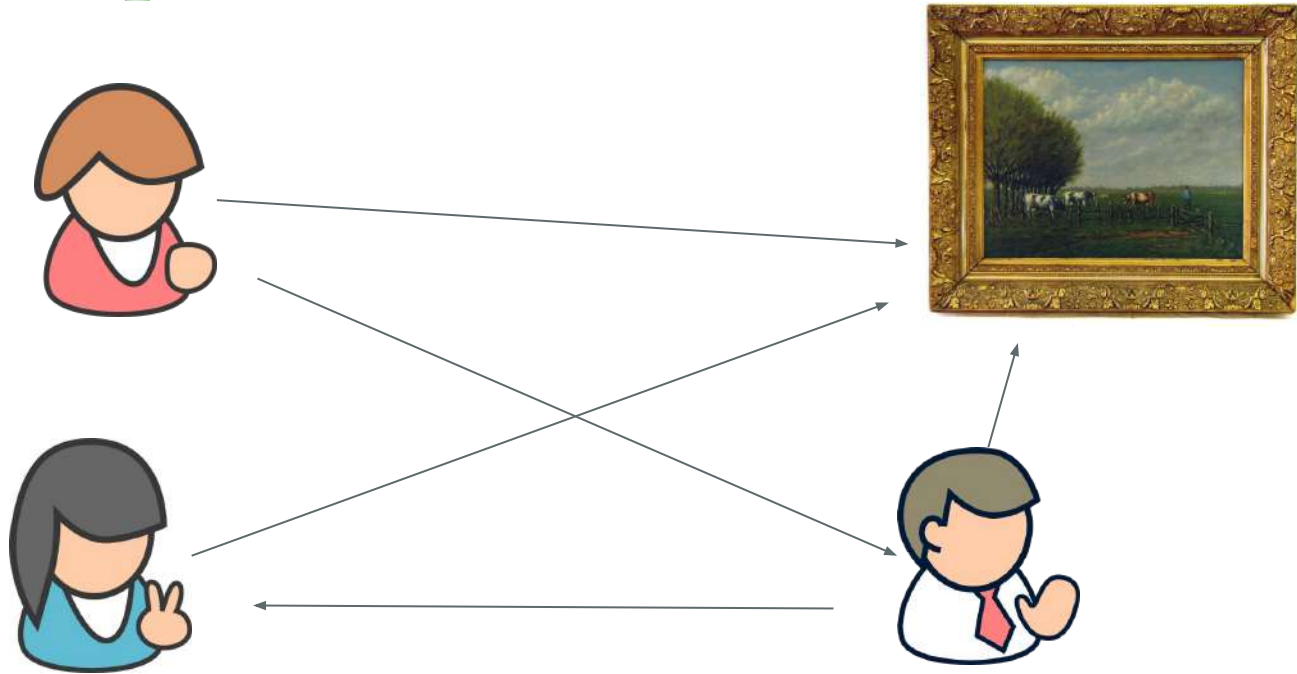
pueden almacenar dinero



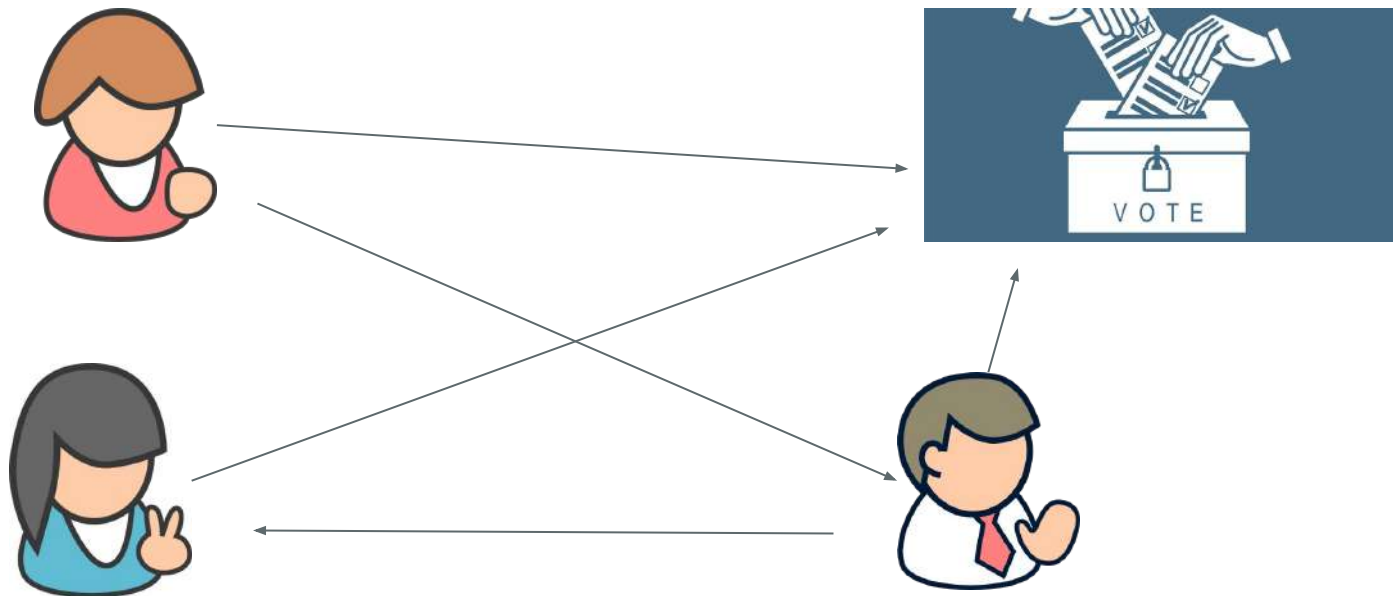
pueden distribuir dinero



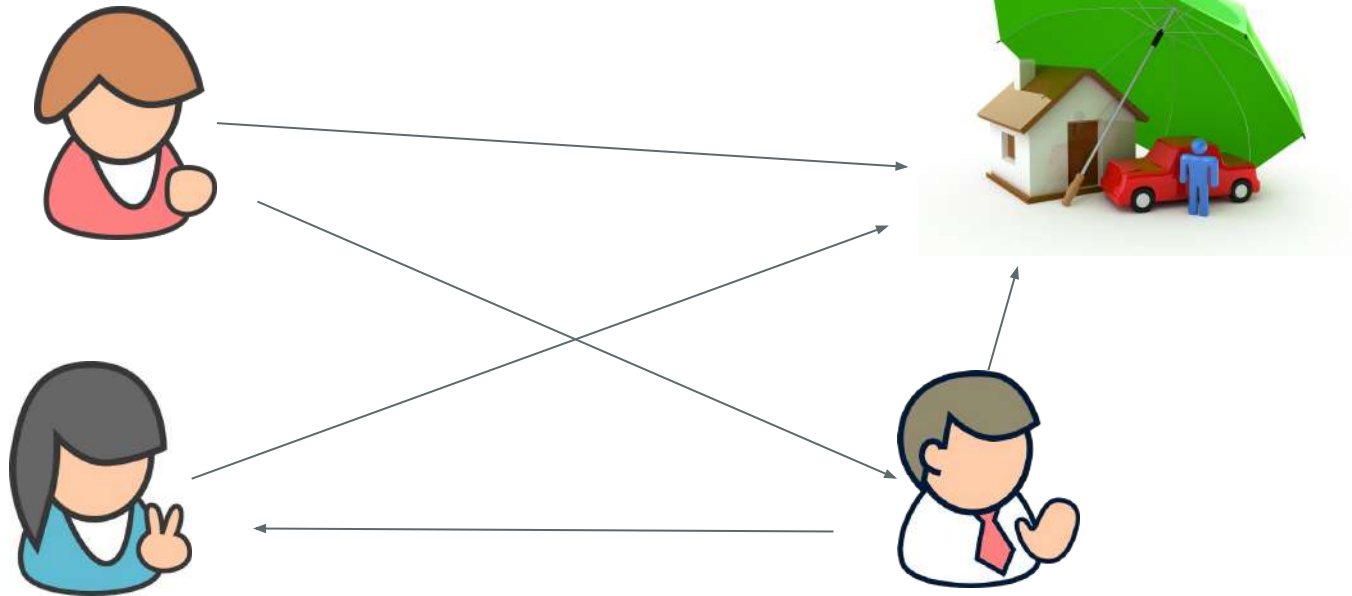
pueden representar obras de arte



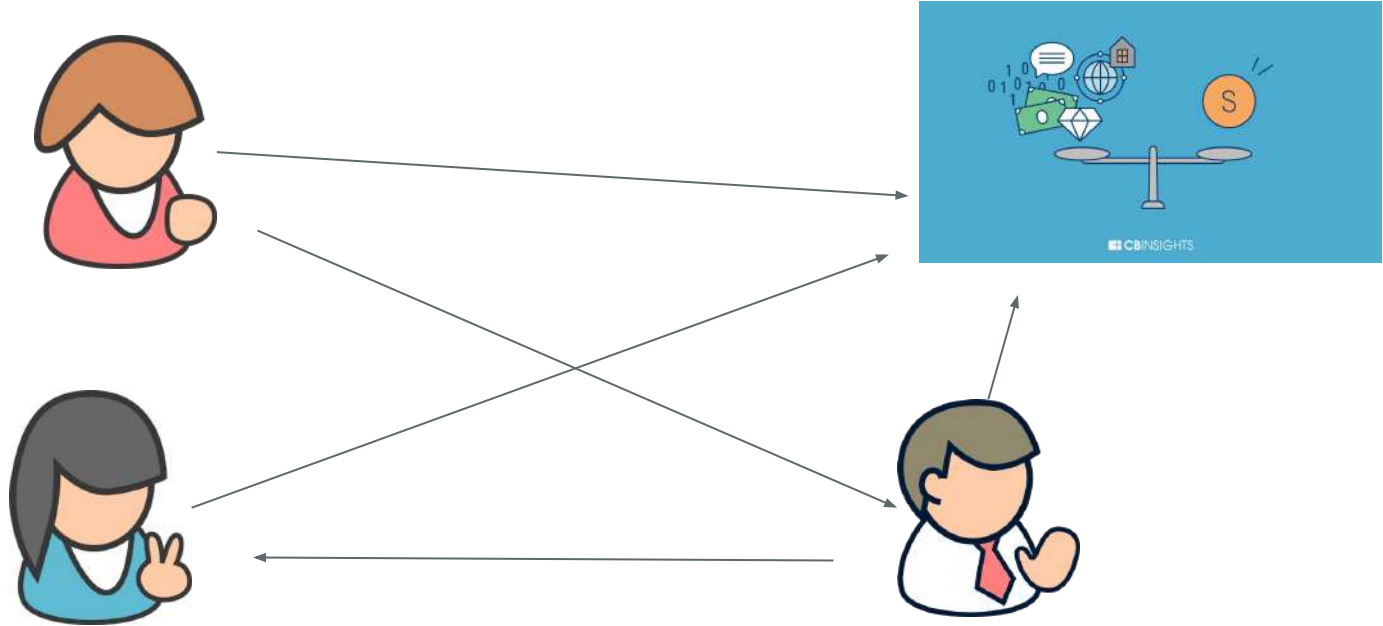
pueden llevar registros de votación



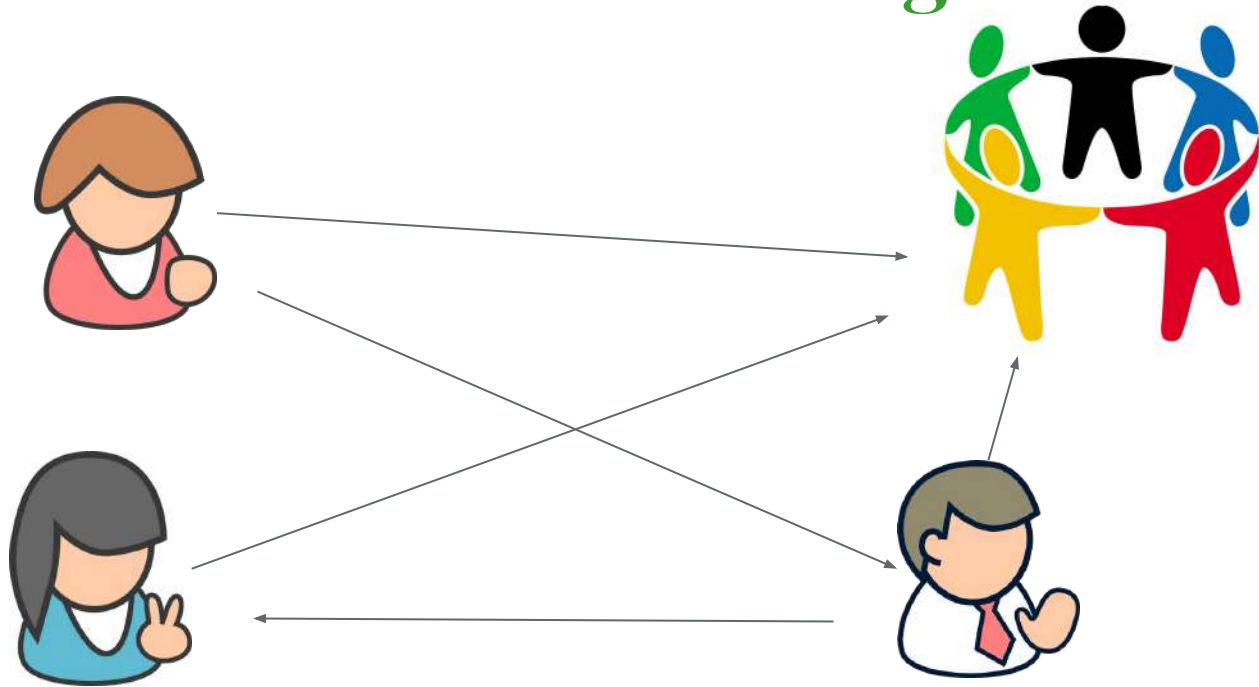
pueden ser pólizas de seguros



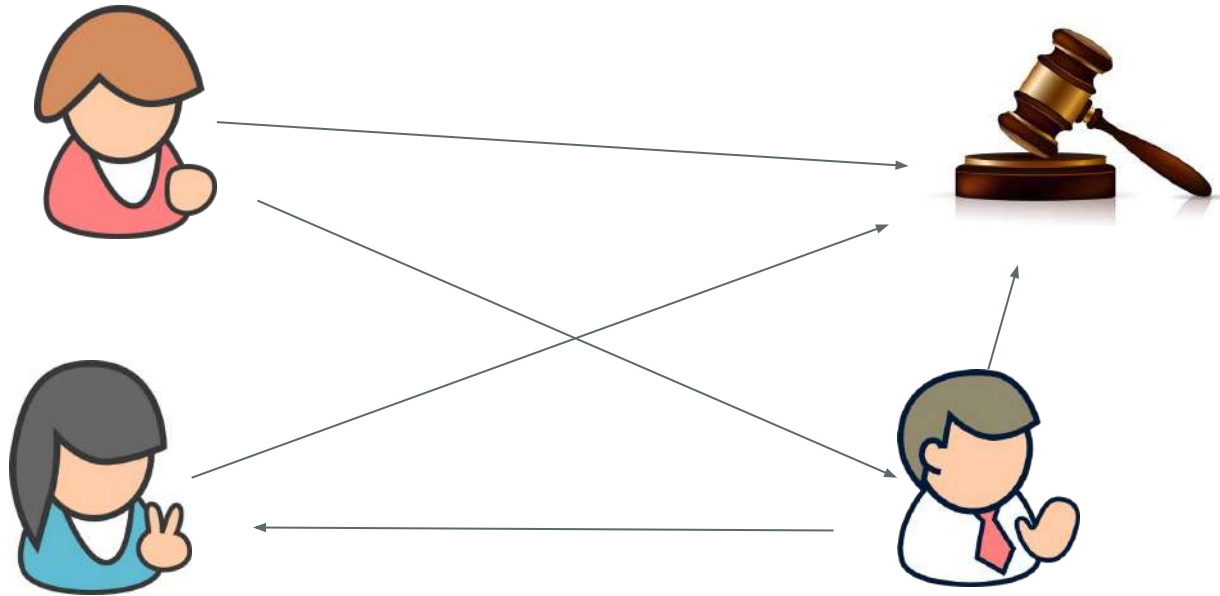
pueden crear monedas estables



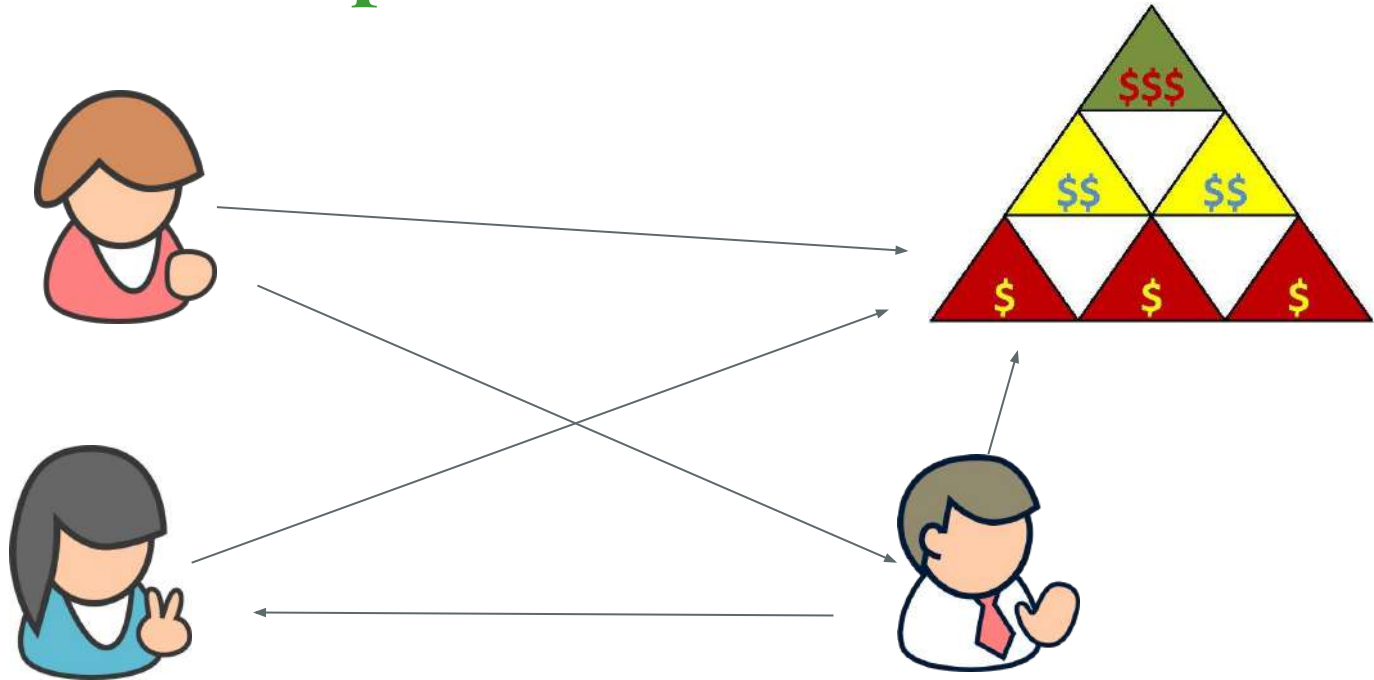
permiten nuevas formas de organizarse



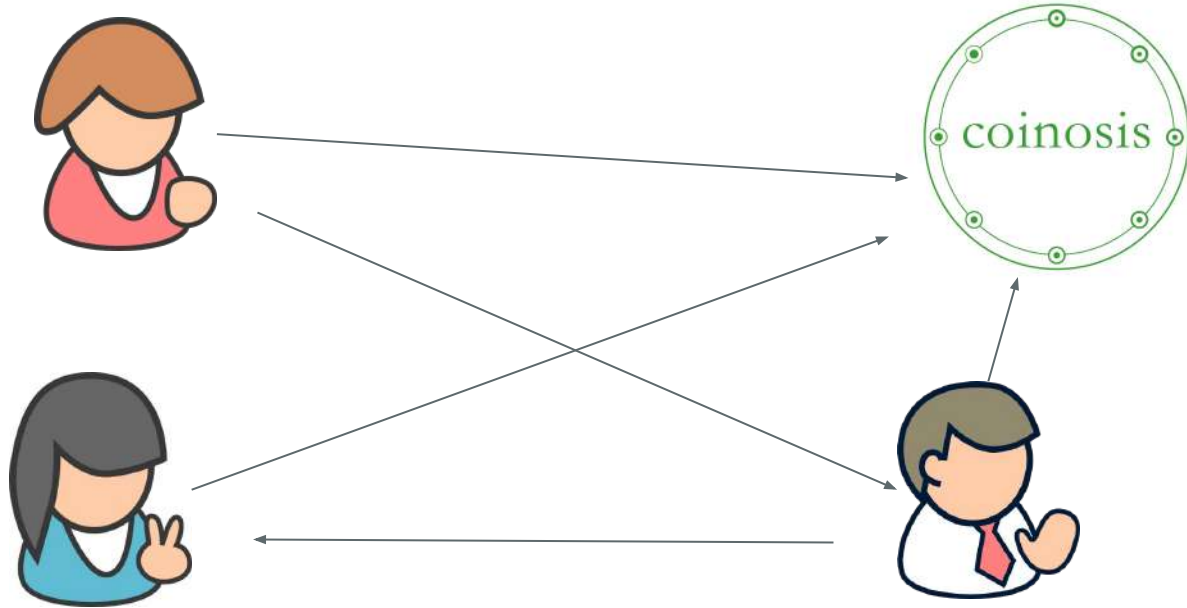
permiten hacer cortes de justicia



permiten hacer pirámides...



y también plataformas educativas



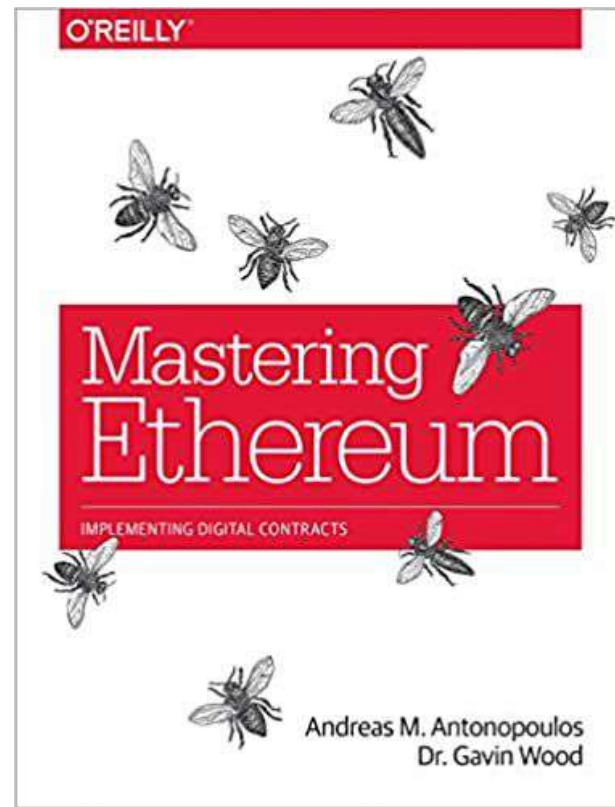
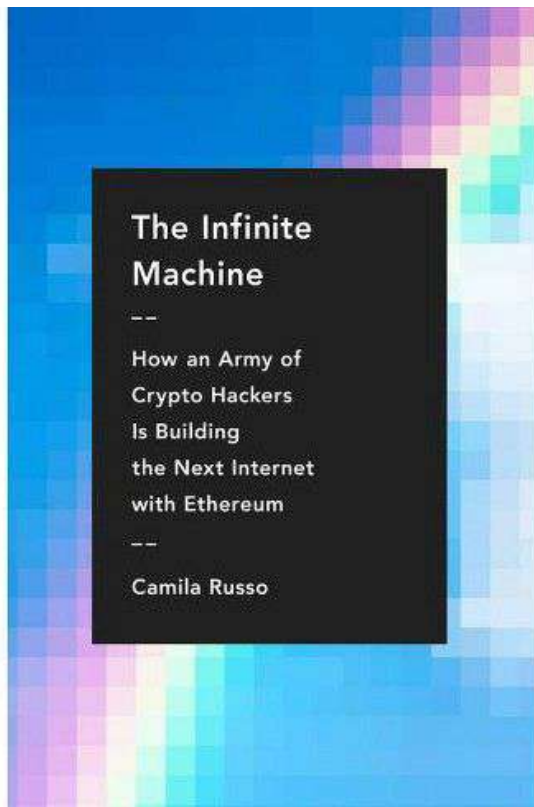


Implicaciones importantes

Con el uso de Ethereum estamos viendo:

- Cambio en la forma de pensar el desarrollo web. Estamos en el Internet del Valor o Web3.0
- Tenemos una herramienta que permite generar redes de confianza entre diversos agentes.
- La privacidad y transparencia son dos valores que están en tensión constante en este nuevo tipo de implementaciones.

Lecturas recomendadas



Índice

0. Conceptos importantes en Ethereum
 - a. Contratos inteligentes
 - b. Descentralización
 - c. Aplicaciones descentralizadas
- 1. Desintermediación de mercados**
2. Diseño de incentivos en el sistema educativo
3. Corrupción en el manejo de recursos públicos
4. Desafíos, lecturas adicionales
5. Sorpresa para Colombia 🦄



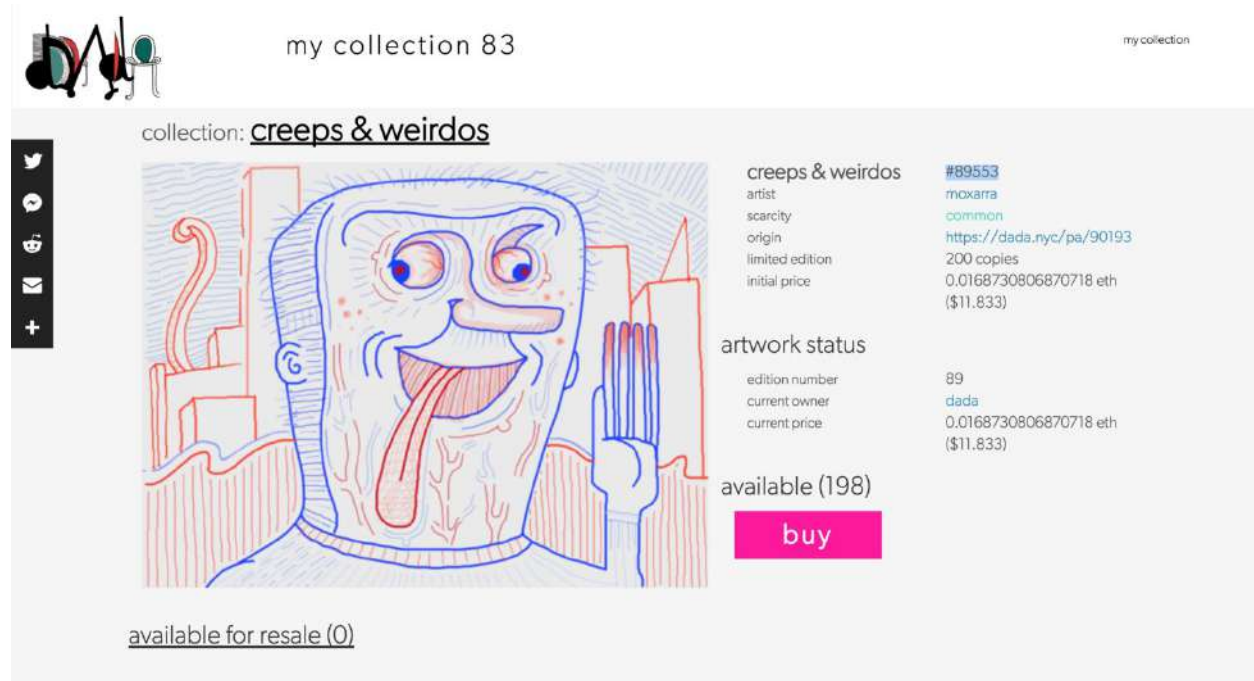
¿Cómo podemos corregir algunas fallas de mercado usando ~~XX~~?

Podemos ver cuatro áreas donde la blockchain puede corregir 4 problemas

- Aumentar las ventas digitales mediante escasez digital.
- Democratizar la inversión en arte.
- Reducir la falsificación de las obras.
- Crear una forma justa para pagarle a los artistas.

5 características para solucionar los 4 problemas

1. Autenticación/Atribución
2. Ediciones digitales
3. Prueba de propiedad
4. Proveniencia/Trazabilidad
5. Remuneración



The screenshot shows an NFT marketplace interface. At the top, there's a navigation bar with a logo on the left, 'my collection 83' in the center, and 'my collection' on the right. Below the navigation bar, the collection name 'creeps & weirdos' is displayed. To the left of the artwork is a vertical social media sharing bar with icons for Twitter, Facebook, and a plus sign. The central artwork is a blue and red line drawing of a character with a long tongue sticking out. To the right of the artwork, there are two columns of metadata. The first column lists 'creeps & weirdos' as the collection name, 'moxarra' as the artist, 'scarcity' as 'common', 'origin' as 'https://dada.nyc/pa/90193', 'limited edition' as '200 copies', and 'initial price' as '0.0168730806870718 eth (\$11.833)'. The second column lists 'artwork status' as '89', 'edition number' as 'dada', 'current owner' as '0.0168730806870718 eth', and 'current price' as '(\$11.833)'. Below the artwork, there's a pink 'buy' button and a label 'available (198)'. At the bottom left, there's a label 'available for resale (0)'.

my collection 83

my collection

collection: creeps & weirdos

creeps & weirdos
artist
scarcity
origin
limited edition
initial price

#89553
moxarra
common
<https://dada.nyc/pa/90193>
200 copies
0.0168730806870718 eth
(\$11.833)

artwork status

edition number
current owner
current price

89
dada
0.0168730806870718 eth
(\$11.833)

available (198)

buy

available for resale (0)

Y la gente sí compra arte digital?



Hasta pixeles

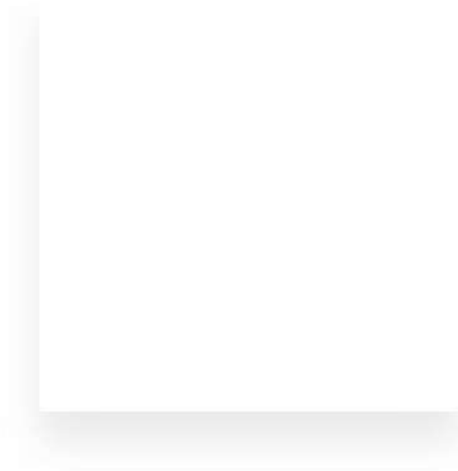
SuperRare

Alpha

Edition 1 of 1

Alpha Pixel
#alpha #pak #pixel
#transparent

SIGN UP TO COLLECT



HISTORY

@museumofcryptoart transferred to @moca
A MONTH AGO [\[view tx\]](#)

@museumofcryptoart bought from @pak for 55.555Ξ (\$15,370)
A MONTH AGO [\[view tx\]](#)

HISTORY

@museumofcryptoart transferred to @moca

A MONTH AGO [\[view tx\]](#)

@museumofcryptoart bought from @pak for 55.555Ξ (\$15,370)

A MONTH AGO [\[view tx\]](#)

@_____ placed a bid of 29.1262Ξ

A MONTH AGO [\[view tx\]](#)

@whaleshark placed a bid of 25.0Ξ

A MONTH AGO [\[view tx\]](#)

<https://superrare.co/artwork-v2/alpha-7713>

Lectura recomendada

Introducing The Invisible Economy

The Invisible Economy is the radical separation of art and the market. Blockchain technology allows the economy to be both invisible and transparent.



DADA.art [Follow](#)
May 17 · 4 min read



Part 1 of 12.

By Beatriz Ramos and Yehudit Mam

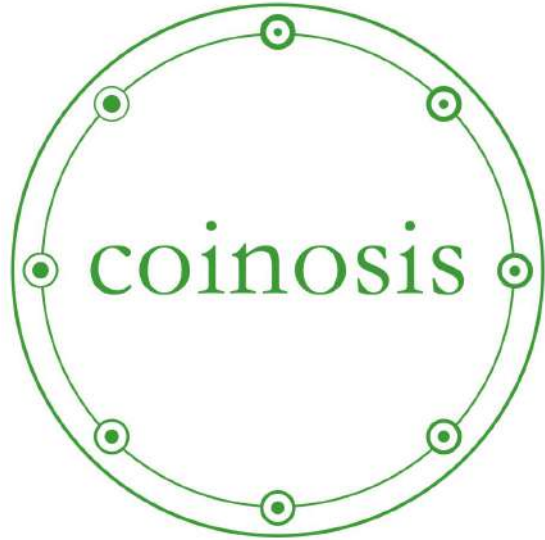


Art by Mhxarra. Owned by Nitrous8.

<https://medium.com/@PowerDada/the-invisible-economy-db46897d4f07>

Índice

0. Conceptos importantes en Ethereum
 - a. Contratos inteligentes
 - b. Descentralización
 - c. Aplicaciones descentralizadas
1. Desintermediación de mercados
2. **Diseño de incentivos en el sistema educativo**
3. Corrupción en el manejo de recursos públicos
4. Desafíos, lecturas adicionales
5. Sorpresa para Colombia 🦄



Cómo funciona
coinosis

¿Qué solucionamos?

Naval (@naval) · Jul 22, 2016
Free education is abundant, all over the Internet. It's the desire to learn that's scarce.
♡ 85 👍 1.5K 💬 1.5K

learn.consensys.net

CRYPTOCURRENCY WALLETS
What is a Crypto Wallet?

FREE

STABLECOINS
Stablecoins

FREE

THE TOKENIZED FUTURE
The Tokenized Future

FREE

Telogram

crypt Ethereum Baybati
79 members

What would you like to hear more this Saturday 22 3:00pm to 4:30pm?

0% What is a Crypto Wallet?
25% Stablecoins
75% The Tokenized Future

12 votes

	CLASS	REWARD
1. Gustavo	72	23.2%
2. Maria	53	18.7%
3. Camila	32	11.2%
⋮	⋮	⋮

Expected Result

COINOSIS

coinsis.co/the-tokenized-future

The Tokenized Future
3/22/2016, 3:07:00 PM

make a deposit of to participate

¿Cómo lo hacemos?

¿Qué estudiaremos?

- Introducir el concepto de tokens y cómo funciona la tokenización en Ethereum.
- Entender cuáles son los beneficios de la tokenización de un activo.
- Enseñarte cómo crear tu propio token usando la blockchain de Ethereum.



Material de estudio:

<https://learn.consensys.net/catalog/info/id:174>

Lugar de encuentro:

<https://coinosis.co/study-group-the-tokenized-future>

Fecha y hora:

Sábado 22 de Agosto
3:00 a 4:30pm



Círculo de Estudio

The Tokenized Future


¿Cómo funciona una sesión en coinosis?

Partimos de este idea: "La educación gratuita es abundante, todo está en Internet. Lo que hace falta es un círculo de estudio."

- 1 Todos ponemos para entrar al grupo de estudio. Será una contribución que va a un contrato inteligente.
- 2 Participas con tu conocimiento y calificas el aporte de los demás. Recuerda revisar previamente el material de estudio.
- 3 Según el reconocimiento de los participantes se reparte proporcionalmente lo depositado.

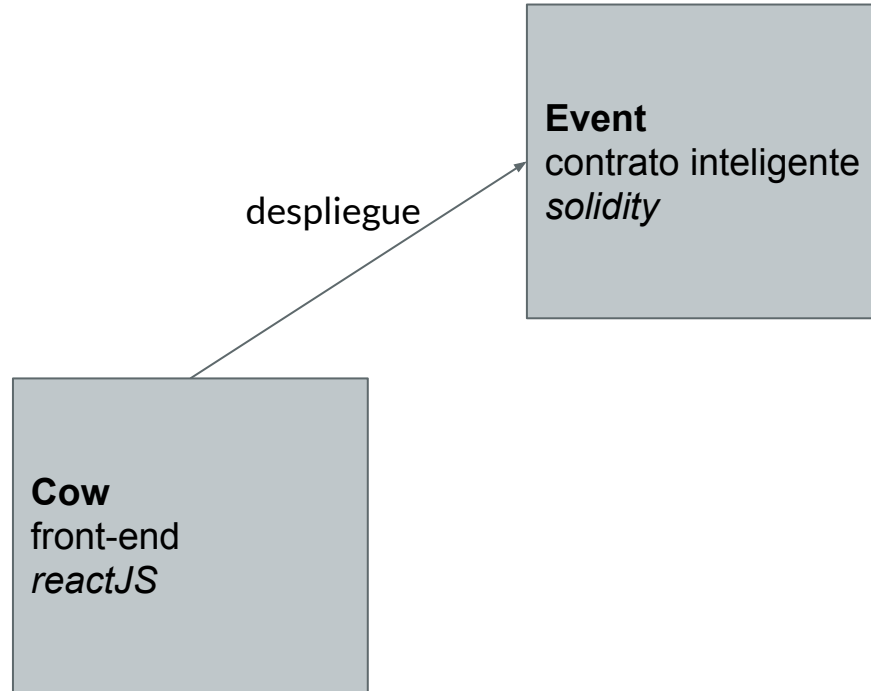
Así funciona nuestra forma de aprender colaborativa y descentralizadamente.

Arquitectura de una dApp

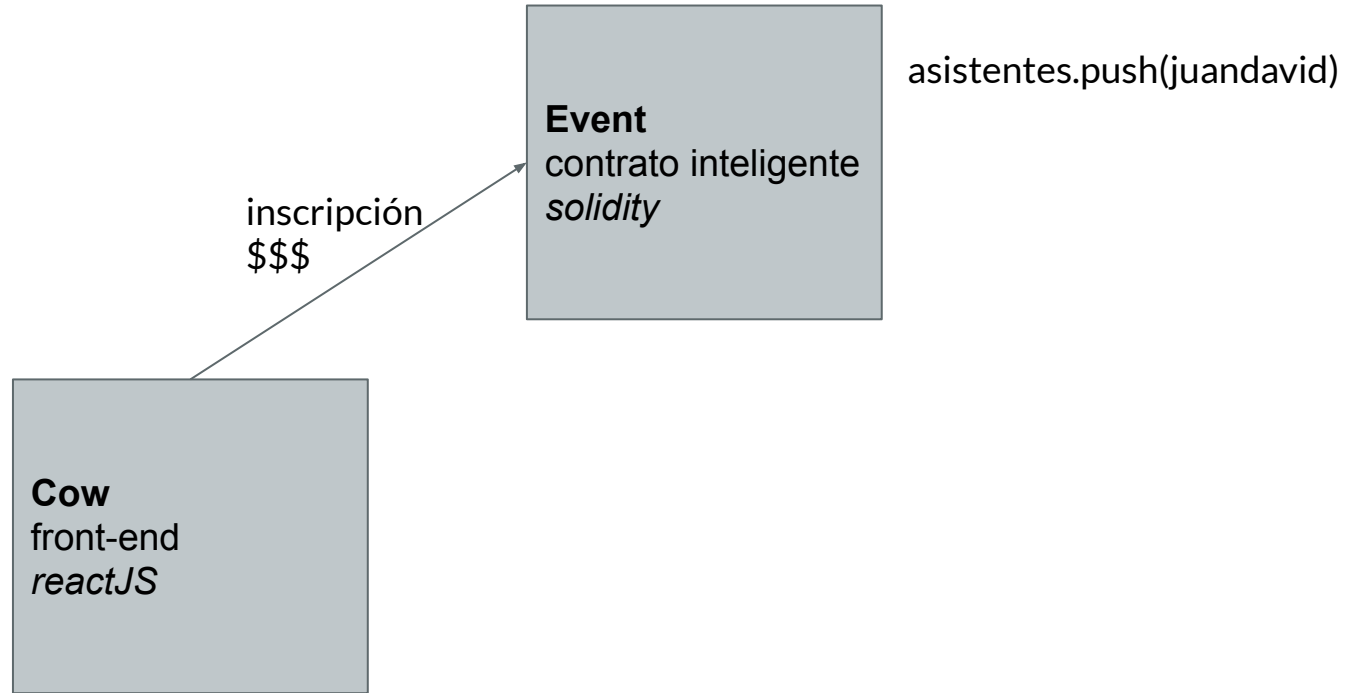


Cow
front-end
reactJS

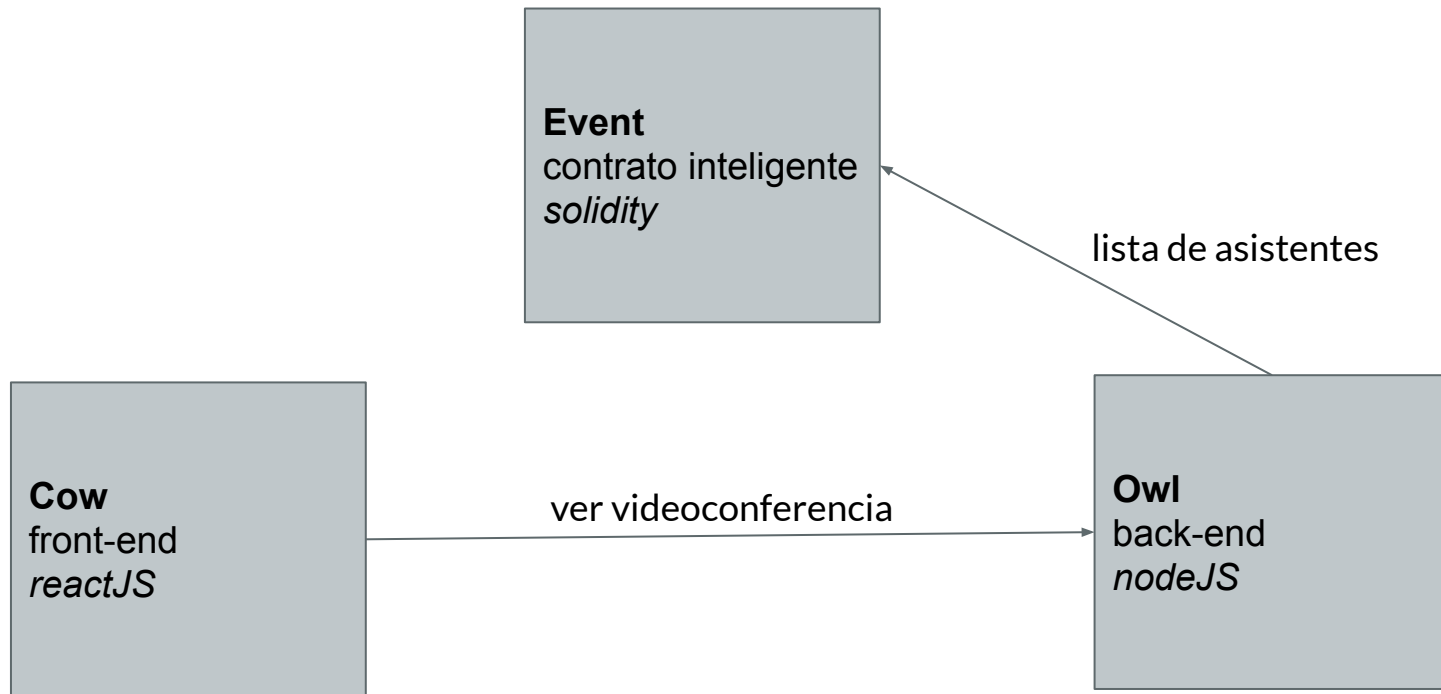
Arquitectura de una dApp



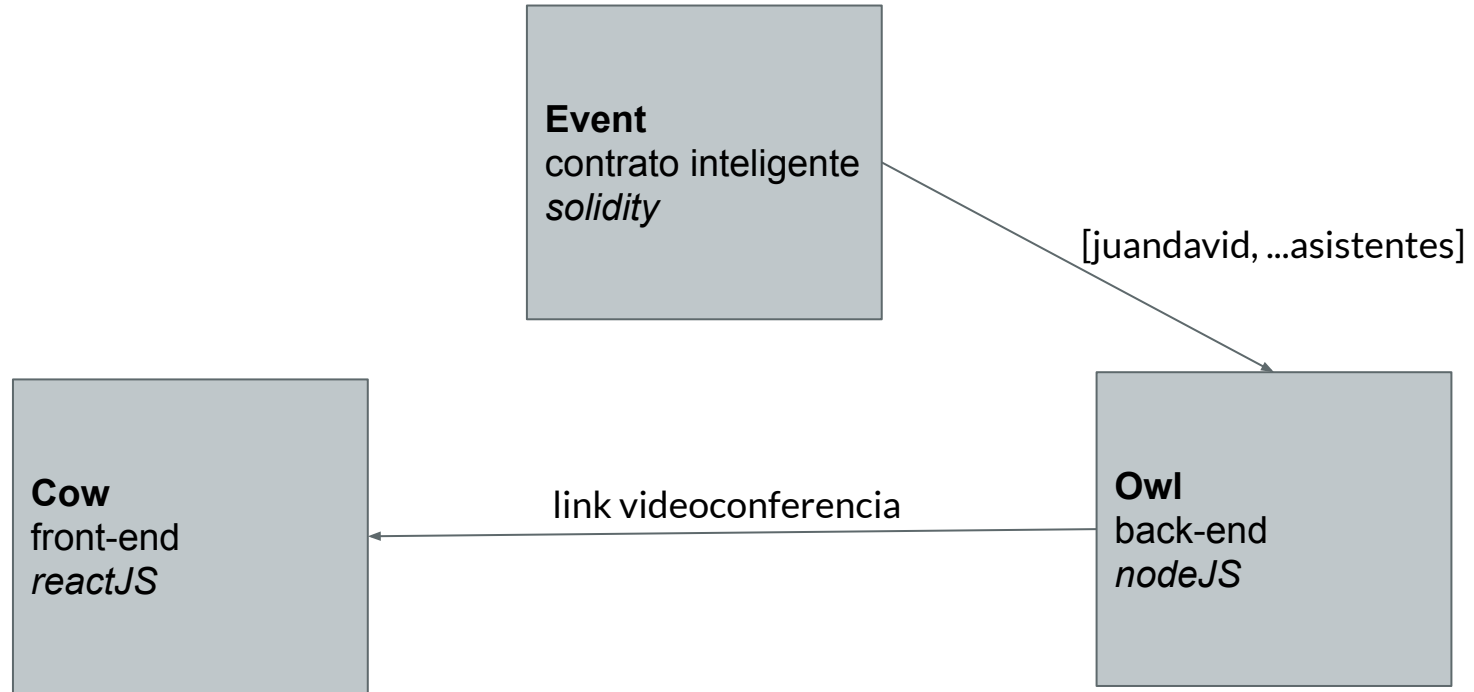
Arquitectura de una dApp



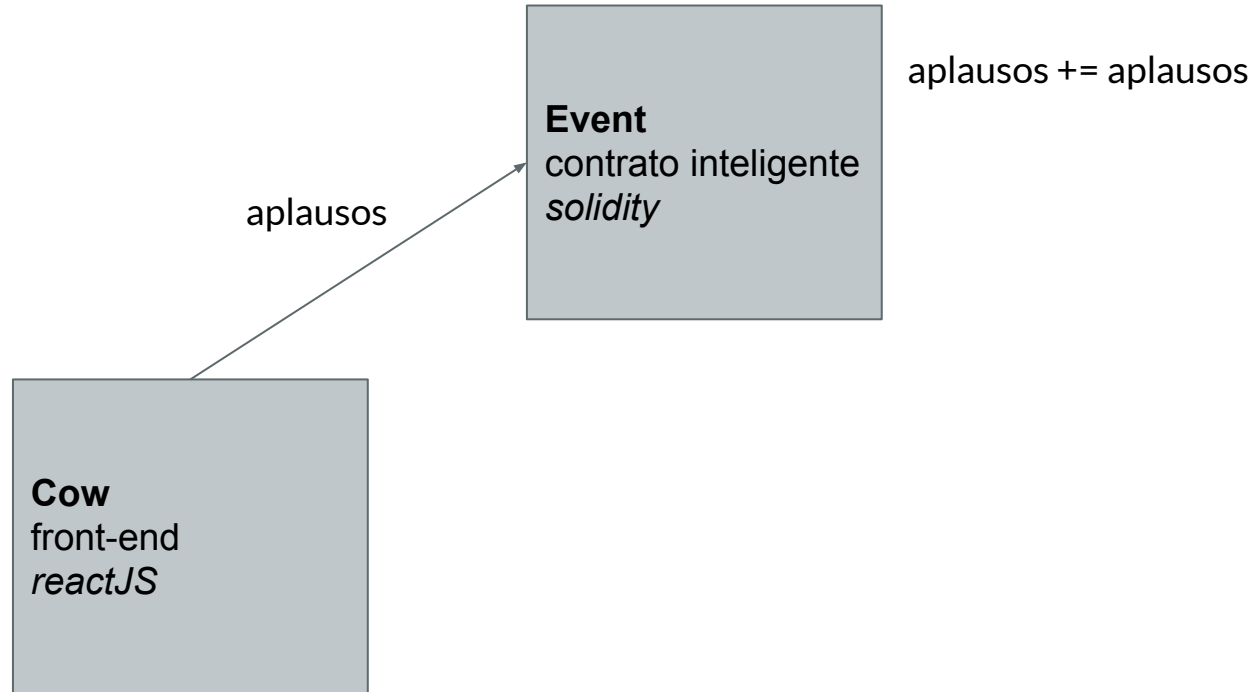
Arquitectura de una dApp



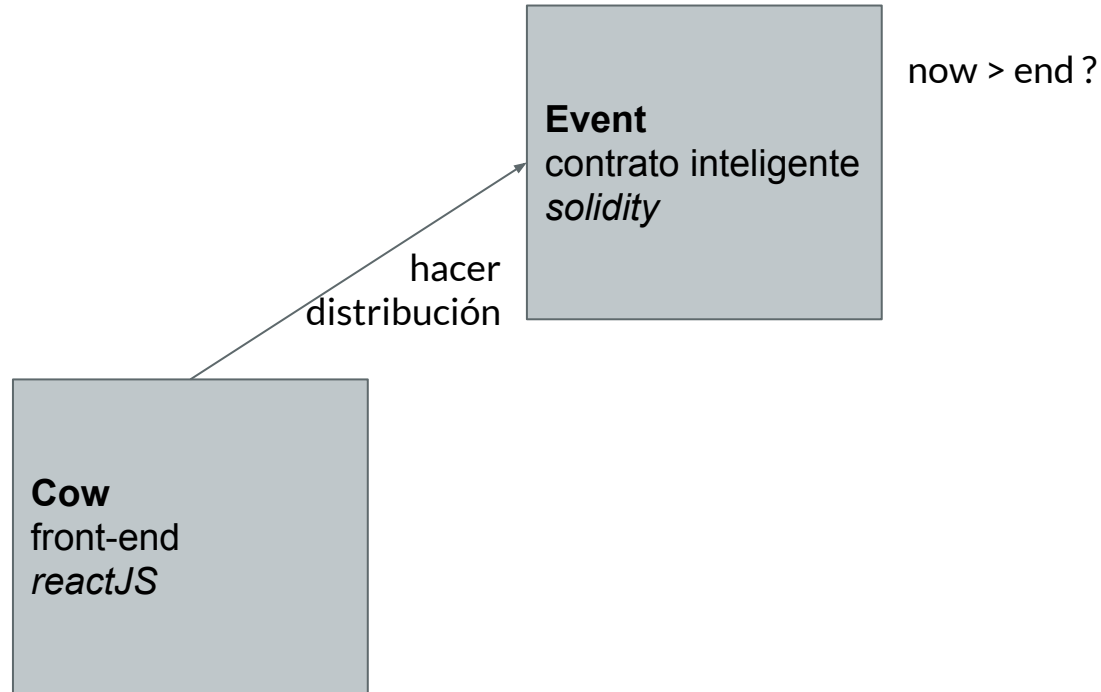
Arquitectura de una dApp



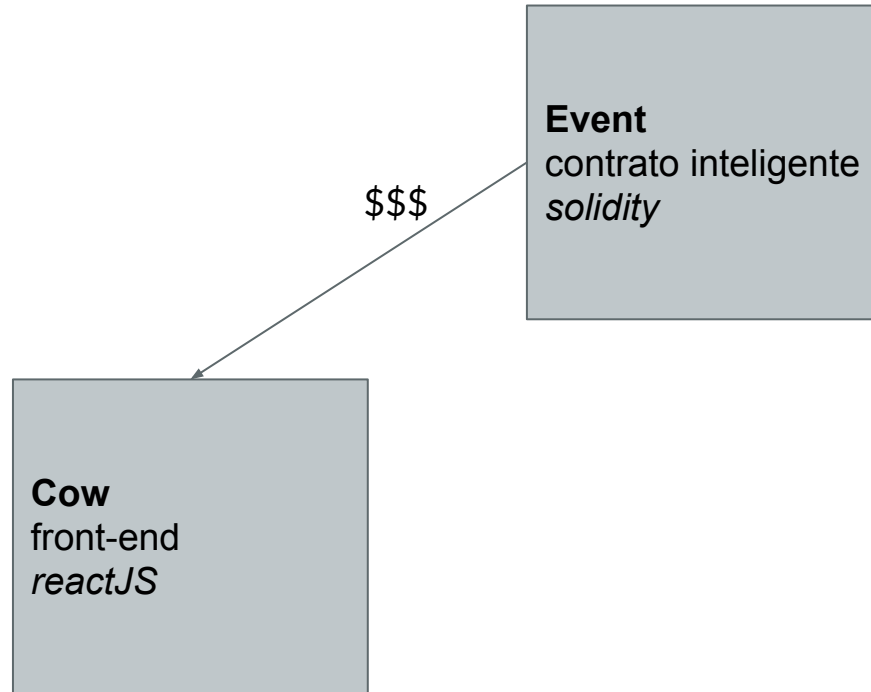
Arquitectura de una dApp



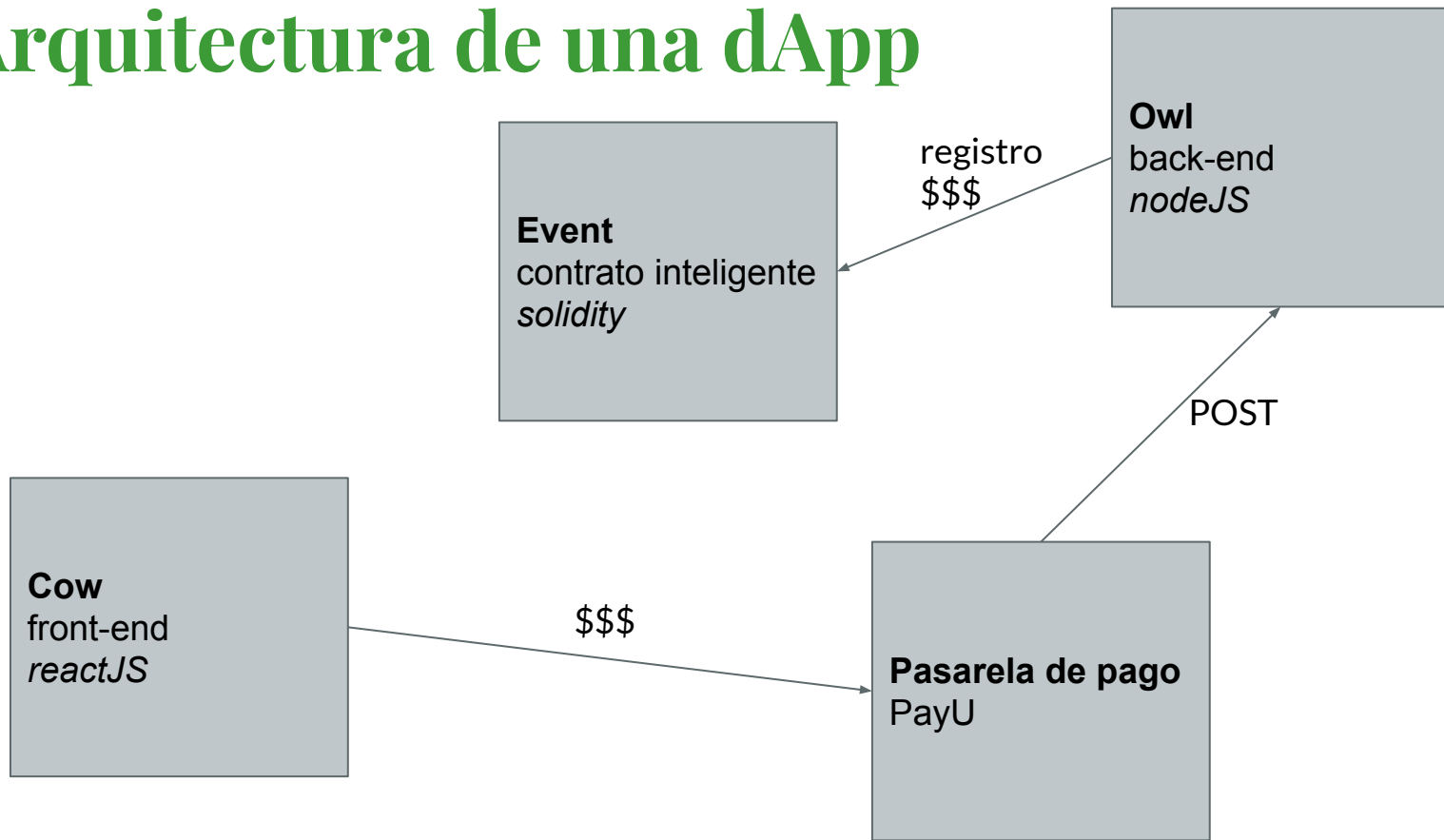
Arquitectura de una dApp



Arquitectura de una dApp

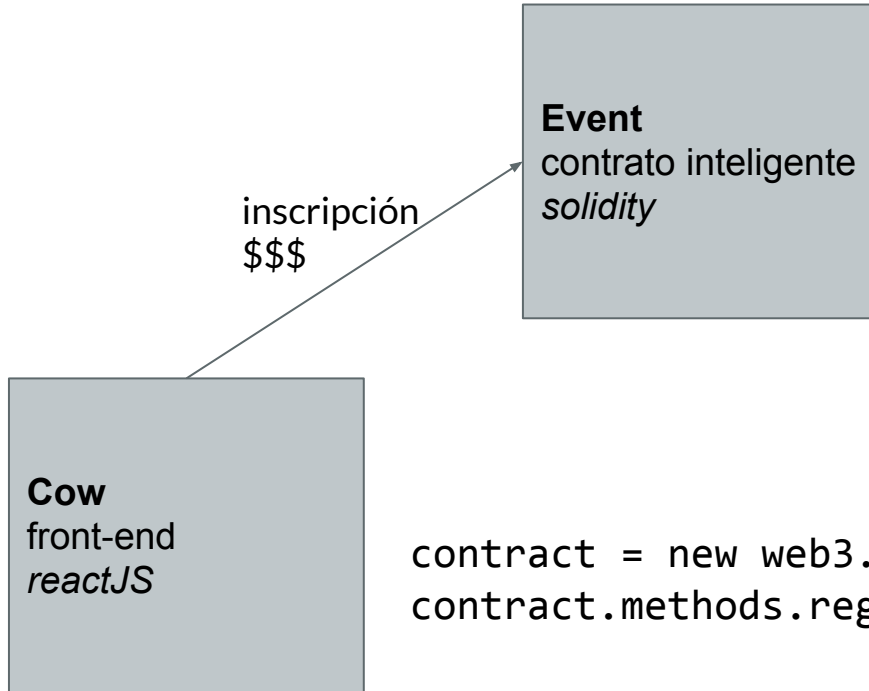


Arquitectura de una dApp



Algo de código

inscripción

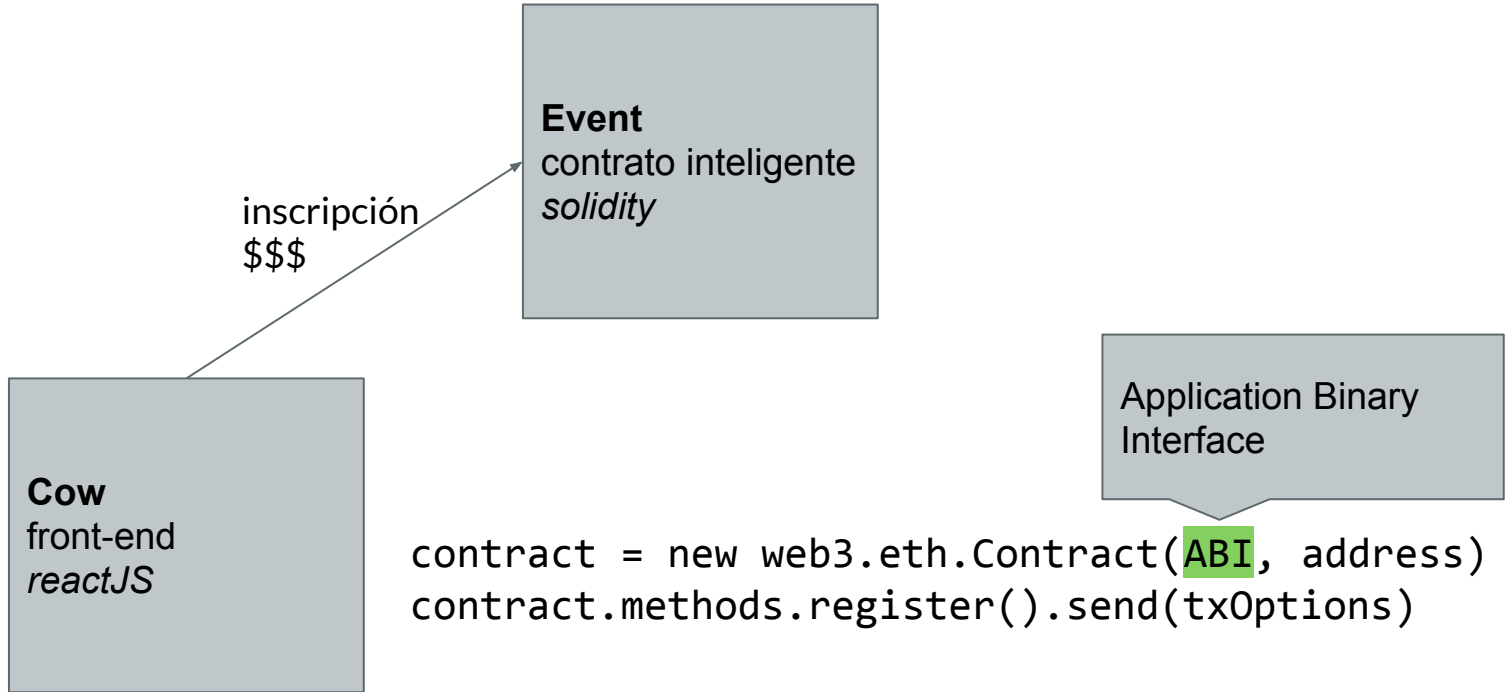


```
contract = new web3.eth.Contract(ABI, address)
contract.methods.register().send(txOptions)
```

inscripción



inscripción



inscripción

Cow
front-end
reactJS

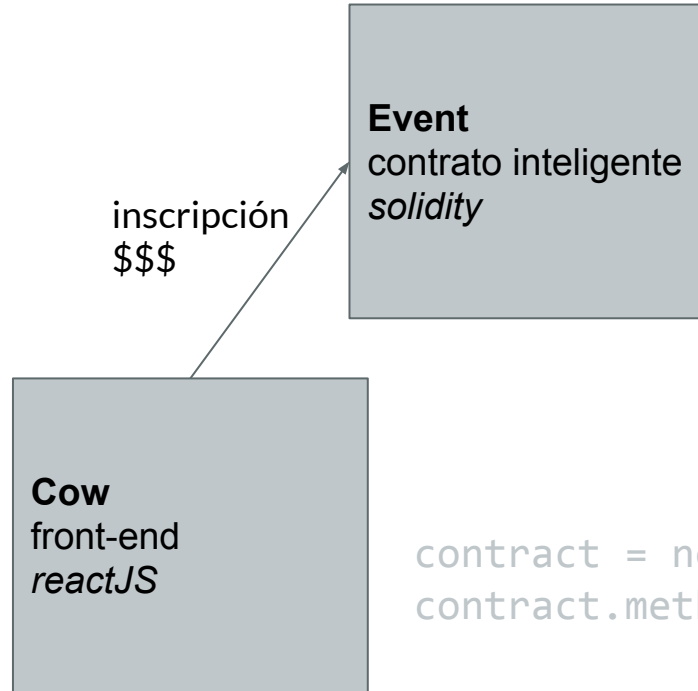
inscripción
\$\$\$

Event
contrato inteligente
solidity

```
contract = new web3.eth.Contract(ABI, address)
contract.methods.register().send(txOptions)
```

```
{
  from: '0xabc123...',
  value: 0.017,
  gasPrice: 0.000001,
}
```

inscripción



```
function register () external payable {  
    register(msg.sender, msg.value);  
}
```

```
contract = new web3.eth.Contract(ABI, address)  
contract.methods.register().send(txOptions)
```



inscripción

```
function register (address payable _attendee, uint256 _fee)
internal {
    require(_fee == fee);
    require(states[_attendee] == ATTENDEE_UNREGISTERED);
    require(attendees.length < MAX_ATTENDEES);
    require(block.timestamp < end);
    states[_attendee] = ATTENDEE_REGISTERED;
    attendees.push(_attendee);
}
```

inscripción

el nombre de
la función

```
function register (address payable _attendee, uint256 _fee)
internal {
    require(_fee == fee);
    require(states[_attendee] == ATTENDEE_UNREGISTERED);
    require(attendees.length < MAX_ATTENDEES);
    require(block.timestamp < end);
    states[_attendee] = ATTENDEE_REGISTERED;
    attendees.push(_attendee);
}
```

inscripción

los argumentos

```
function register (address payable _attendee, uint256 _fee)
internal {
    require(_fee == fee);
    require(states[_attendee] == ATTENDEE_UNREGISTERED);
    require(attendees.length < MAX_ATTENDEES);
    require(block.timestamp < end);
    states[_attendee] = ATTENDEE_REGISTERED;
    attendees.push(_attendee);
}
```

inscripción

la dirección de la
persona que se está
registrando

```
function register (address payable _attendee, uint256 _fee)
internal {
    require(_fee == fee);
    require(states[_attendee] == ATTENDEE_UNREGISTERED);
    require(attendees.length < MAX_ATTENDEES);
    require(block.timestamp < end);
    states[_attendee] = ATTENDEE_REGISTERED;
    attendees.push(_attendee);
}
```

inscripción

dirección pagable,
para poderle pagar
en el futuro

```
function register (address payable _attendee, uint256 _fee)
internal {
    require(_fee == fee);
    require(states[_attendee] == ATTENDEE_UNREGISTERED);
    require(attendees.length < MAX_ATTENDEES);
    require(block.timestamp < end);
    states[_attendee] = ATTENDEE_REGISTERED;
    attendees.push(_attendee);
}
```

inscripción

la cantidad de dinero
recibido

```
function register (address payable _attendee, uint256 _fee)
internal {
    require(_fee == fee);
    require(states[_attendee] == ATTENDEE_UNREGISTERED);
    require(attendees.length < MAX_ATTENDEES);
    require(block.timestamp < end);
    states[_attendee] = ATTENDEE_REGISTERED;
    attendees.push(_attendee);
}
```


inscripción

unsigned integer de
256 bits = 32 bytes =
 2^{256}

```
function register (address payable _attendee, uint256 _fee)
internal {
    require(_fee == fee);
    require(states[_attendee] == ATTENDEE_UNREGISTERED);
    require(attendees.length < MAX_ATTENDEES);
    require(block.timestamp < end);
    states[_attendee] = ATTENDEE_REGISTERED;
    attendees.push(_attendee);
}
```

inscripción

la visibilidad

```
function register (address payable _attendee, uint256 _fee)
internal {
    require(_fee == fee);
    require(states[_attendee] == ATTENDEE_UNREGISTERED);
    require(attendees.length < MAX_ATTENDEES);
    require(block.timestamp < end);
    states[_attendee] = ATTENDEE_REGISTERED;
    attendees.push(_attendee);
}
```

inscripción

seguimos el patrón
checks-effects-interactions

```
function registerAttendee(uint256 _attendee, uint256 _fee)
internal {
    require(_fee == fee);
    require(states[_attendee] == ATTENDEE_UNREGISTERED);
    require(attendees.length < MAX_ATTENDEES);
    require(block.timestamp < end);
    states[_attendee] = ATTENDEE_REGISTERED;
    attendees.push(_attendee);
}
```

inscripción

```
function register (address payable _attendee, uint256 _fee)
internal {
    require(_fee == fee);
    require(states[_attendee] == ATTENDEE_UNREGISTERED);
    require(attendees.length < MAX_ATTENDEES);
    require(block.timestamp < end);
    states[_attendee] = ATTENDEE_REGISTERED;
    attendees.push(_attendee);
}
```

checks

inscripción

```
function register (address payable _attendee, uint256 _fee)
internal {
    require(_fee == fee);
    require(states[_attendee] == ATTENDEE_UNREGISTERED);
    require(attendees.length < MAX_ATTENDEES);
    require(block.timestamp < end);
    states[_attendee] = ATTENDEE_REGISTERED,
    attendees.push(_attendee);
}
```

effects

inscripción

el dinero enviado tiene
que ser igual al aporte
requerido

```
function payable _attendee, uint256 _fee)
internal {
    require(_fee == fee);
    require(states[_attendee] == ATTENDEE_UNREGISTERED);
    require(attendees.length < MAX_ATTENDEES);
    require(block.timestamp < end);
    states[_attendee] = ATTENDEE_REGISTERED;
    attendees.push(_attendee);
}
```

inscripción

```
function register (address _address, uint256 _fee)
internal {
    require(_fee == fee);
    require(states[_attendee] == ATTENDEE_UNREGISTERED);
    require(attendees.length < MAX_ATTENDEES);
    require(block.timestamp < end);
    states[_attendee] = ATTENDEE_REGISTERED;
    attendees.push(_attendee);
}
```

el participante no
puede haberse
registrado ya

inscripción

```
function register (address _addr, uint256 _fee)
internal {
    require(_fee == fee);
    require(states[_attendee] == ATTENDEE_UNREGISTERED);
    require(attendees.length < MAX_ATTENDEES);
    require(block.timestamp < end);
    states[_attendee] = ATTENDEE_REGISTERED;
    attendees.push(_attendee);
}
```

no puede haber más de 100 asistentes

inscripción

```
function register(_attendee, uint256 _fee)
internal {
    require(_fee > 0);
    require(state[_attendee] == ATTENDEE_UNREGISTERED);
    require(attendees.length < MAX_ATTENDEES);
    require(block.timestamp < end);
    states[_attendee] = ATTENDEE_REGISTERED;
    attendees.push(_attendee);
}
```

el evento no puede haber
finalizado

inscripción

```
function register (address payable _attendee, uint256 _fee)
internal {
    require(_fee > 0);
    require(state[_attendee] == ATTENDEE_UNREGISTERED);
    require(attendees[_attendee] == 0);
    require(block.timestamp < end);
    states[_attendee] = ATTENDEE_REGISTERED;
    attendees.push(_attendee);
}
```

el usuario queda en estado
registrado

inscripción

```
function register (address payable _attendeer, uint256 _fee)
internal {
    require(_fee == fee);
    require(attendees[_attendeer] == ATTENDEE_UNREGISTERED);
    require(attendees.length < MAX_ATTENDEES);
    require(_attendeer < end);
    states[_attendeer] = ATTENDEE_REGISTERED;
    attendees.push(_attendeer);
}
```

se agrega a la lista de asistentes

Distribución



Juan David

[← atrás](#)

[Círculo de Estudio] The Tokenized Future

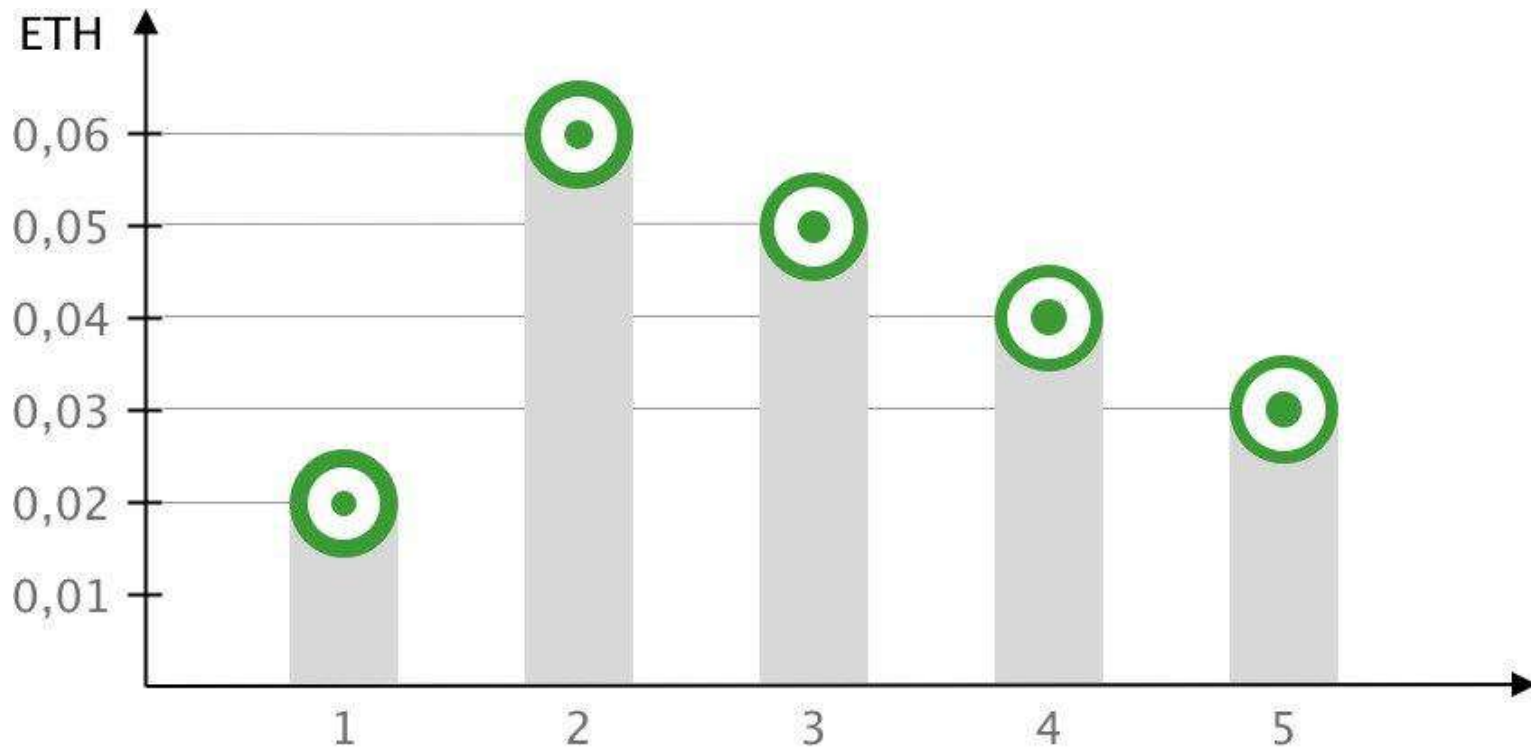
8/22/2020, 3:00:00 PM

distribución 0xfc6b...5b04 8/22/2020, 4:31:45 PM (8 participantes)

aporte por persona: 2.00 USD aporte total: 16.00 USD

participante	aplausos	porcentaje	recompensa	balance	estado
<u>Juan David</u>	136	22.4 %	3.58 USD	1.58 USD	<u>enviada</u>
<u>Andres Montes</u>	30	4.9 %	0.79 USD	-1.21 USD	<u>enviada</u>
<u>Emilio Silva</u>	112	18.4 %	2.95 USD	0.95 USD	<u>enviada</u>
<u>Dilver</u>	30	4.9 %	0.79 USD	-1.21 USD	<u>enviada</u>
<u>Gustavo</u>	141	23.2 %	3.71 USD	1.71 USD	<u>enviada</u>
<u>Juan Cuenca Vega</u>	40	6.6 %	1.05 USD	-0.95 USD	<u>enviada</u>
<u>ErrollC</u>	91	15.0 %	2.39 USD	0.39 USD	<u>enviada</u>
<u>Luis Herrera PA</u>	28	4.6 %	0.74 USD	-1.26 USD	<u>enviada</u>
total	608	100.0 %	16.00 USD	-0.00 USD	

Escala de contribuciones



Total: 0,2 ETH / 171.029 COP 🇨🇴 / 6.600 ARS 🇲🇯 / 1.066 MXN 🇲🇪

Índice

0. Conceptos importantes en Ethereum
 - a. Contratos inteligentes
 - b. Descentralización
 - c. Aplicaciones descentralizadas
1. Desintermediación de mercados
2. Diseño de incentivos en el sistema educativo
3. **Corrupción en el manejo de recursos públicos**
4. Desafíos, lecturas adicionales
5. Sorpresa para Colombia 🦄

TENDENCIAS > CLARENTENA EN COLOMBIA CORONAVIRUS EN BOGOTÁ ÁLVARO URIBE VÉLEZ EPM MASACRES EN COLOMBIA

VER MÁS

INFORME | 8/14/2018 12:05:00 PM



84.000 millones de pesos, lo que valdrían las irregularidades en alimentación escolar

El contralor general, Edgardo Maya Villazón, reveló que por este tema hay 154 procesos avanzados que involucran

EL TIEMPO



SUSCRIBIRME

INICIAR SESIÓN

JUSTICIA | CORTES | CONFLICTO Y NARCOTRÁFICO | INVESTIGACIÓN | DELITOS | SERVICIO | JURISDICCIÓN ESPECIAL PAZ

El saqueo que produjo el revolcón al Programa de Alimentación Escolar

Tamales facturados por empresas de grúas y pechugas vendidas por ferreterías, entre los escándalos.



Compartir



48 Comentar



Guardar



Reportar



Procurador presentó proyecto que incorpora tecnología blockchain en la vigilancia a la contratación pública

PÚBLICAS UTILIZANDO TECNOLOGÍA BLOCKCHAIN



Ximena Lombana - Gerente Unidad Ejecutora BID-PGN

Fernando Carrillo, Procurador General de la Nación

Dolly Montoya / Rectora UNAL

Mauricio Tovar Gutiérrez, Co-director InTIColombia

Ashley Lannquist-World Economic Forum

WORLD

Problem: Corruption in public procurement

- Government contracting or tendering
- Governments at local, regional, national levels purchase goods, services, and other works (e.g. building bridges and airports, to supplying schools and hospitals)
- \$9.5 trillion spent by governments on procurement contracts annually

Corruption in public procurement is staggering

- 57% of foreign bribery cases are for obtaining a public procurement contract
- 10-30% of public contract's overall value lost to corruption, diverted to pockets of corrupt officials (OECD, UNODC)
- Colombia: \$17B annual cost of corruption (5.3% of GDP)



Initiative: Increase government transparency with blockchain-based e-procurement



Partnership to research government transparency with blockchain:

- World Economic Forum
- Inter-American Development Bank
- Office of the Inspector General of Colombia



Process: 4 key phases of public procurement

Each phase presents challenges and opportunities for corruption:

- Undue influence on needs assessments
- Bid offer tailoring
- Bribery
- Vendor bid price collusion
- Conflicts of interest
- Fraudulent submissions and bid evaluations
- Poor contract price “benchmarking”
- In adequate record-keeping



Transparency International, 201

Use case: Procurement of the Colombian public-school meals program

- Provides breakfast and lunch for the most in-need children of Colombia
- Comptroller General revealed disturbing irregularities in pricing and delivery
 - Contractors purchasing chicken breasts at 4x market price
 - 32 million meals going undelivered in 2016



Why blockchain for public procurement?

Improve the vendor bidding and selection phases of public procurement through:

- Permanent and tamper-evident record-keeping
- Real-time procedural transparency and auditability
- Automating bid-evaluation with “smart contracts”
- Anonymous and uncensored citizen engagement



Lectura recomendada

<https://www.weforum.org/reports/exploring-blockchain-technology-for-government-transparency-to-reduce-e-corruption>



COMMITTED TO
IMPROVING THE STATE
OF THE WORLD

Insight Report

Exploring Blockchain Technology for Government Transparency: Blockchain-Based Public Procurement to Reduce Corruption



Índice

0. Conceptos importantes en Ethereum
 - a. Contratos inteligentes
 - b. Descentralización
 - c. Aplicaciones descentralizadas
1. Desintermediación de mercados
2. Diseño de incentivos en el sistema educativo
3. Corrupción en el manejo de recursos públicos
4. **Desafíos, lecturas adicionales**
5. Sorpresa para Colombia 🦄

Desafíos y más información

- Escalabilidad
- Precios por transacción
- Educación financiera: Pirámides para ser parte de un contrato inteligente



<https://t.me/EthereumBogota>



<https://t.me/EthereumCaribe>

Índice

0. Conceptos importantes en Ethereum
 - a. Contratos inteligentes
 - b. Descentralización
 - c. Aplicaciones descentralizadas
1. Desintermediación de mercados
2. Diseño de incentivos en el sistema educativo
3. Corrupción en el manejo de recursos públicos
4. Desafíos, lecturas adicionales
5. Sorpresa para Colombia 🦄

Devcon 6: Hacia Colombia en 2021

<https://blog.ethereum.org/2020/05/28/devcon-hacia-colombia-en-2021/>



Juan David Reyes

Co-fundador <https://coinosis.co/>

@jdreyespaez (twitter y telegram)

jd.reyespaez@gmail.com