EOS – Una Introducción

Ian Grigg
Traducido al español por María A. Maegli y Ana I. Herrerías
con el apoyo de eosMeso – julio 2018

***Resumen* – Las tecnologías actuales para *blockchain* (cadena de bloques) no logran cumplir todo aquello que los desarrolladores y los usuarios finales requieren para establecer de forma satisfactoria contratos que les permita construir negocios de gran escala. Proponemos EOS, una *blockchain* basada en rendimiento y en auto-gobernanza que brinda un sistema operativo para construir aplicaciones distribuidas de gran escala orientadas al consumidor. El presente documento describe el contexto, la visión y la arquitectura del *software* del sistema EOS que estamos construyendo para servir a un amplio y diverso grupo de usuarios con negocios inteligentes.**

Palabras clave – EOS, blockchain, contrato inteligente.

I. INTRODUCCIÓN

Desde hace tiempo se conocen los conceptos de dinero digital *(digital cash)* y contratos inteligentes (*smart contracts*). Sin embargo, hasta hace poco realmente se empezó a dar grandes pasos para implementar ambos conceptos a nuestra realidad.

Este documento describe de forma introductoria el sistema EOS.IO, el *software* que funciona como la base de EOS: una plataforma nueva de valoración general y de contratación. EOS es un sistema que surge en un contexto dominado por tres plataformas de *blockchain*: Bitcoin, Ethereum y Corda. En este texto, se presenta una descripción de EOS y una comparación con las tres *blockchain* previamente mencionadas, debido a que (a) estas abarcan la mayoría de las discusiones en el ámbito de las Tecnologías de Libro Mayor Distribuido (DLT, *Distributed Ledger Technologies*), (b) porque son suficientemente grandes para ser tomados en cuenta, y (c) porque son conocidos por el autor.

Bitcoin (Nakamoto, 2008) parecía ser la palabra clave en una *blockchain* que ofrecía promesas tanto para el dinero digital como para los contratos inteligentes. Aunque cautivó la atención de los *cypherpunks*, de los medios y de los *hodlers,* Bitcoin fracasó en dejar una marca en el ámbito de los negocios. Ethereum (Woods, 2014) quiso cumplir con la promesa de un contrato inteligente al emplear una "computadora global imparable", mientras que Bitshares (Larimer et al., 2014) intentó abrir el mercado para bienes comerciables. Cientos de *blockchains* alternativas de Bitcoin, o *altcoins,* intentaron hacer algunas modificaciones que en realidad quedaron cortas y no lograron hacer un verdadero cambio significativo. Corda (Brown et al., 2016) se retiró completamente de *blockchain* y exploró soluciones de flujo de trabajo entre una parte y otra (*party-to-party workflow solutions)*.

Aunque parece que ya estamos muy cerca, los usuarios finales aún no están del todo satisfechos. Por lo tanto, estamos a tiempo de ver de nuevo, desde el punto de vista del usuario final, ¿para qué es la demanda? Así, se podrá

Ian Grigg es un criptógrafo financiero y socio de block.one. (ver http//iang.org/). Aclaraciones:

(i) Este documento explica en términos generales el *software* EOS.IO, el cual le permite a cualquier comunidad iniciar una *blockchain* EOS. Puesto que el *software* es de entrada abierta y cualquier comunidad es libre de ejercer cualquier control sobre la misma siempre y cuando no esté limitado por su Constitución, este documento podría indicar, mas no aseverar respecto de alguna *blockchain* EOS que se desee iniciar.

(ii) He tratado que este documento sea lo más imparcial posible, pero las opiniones y los sesgos son inevitables y son lo que hacen que esta vida sea tan especial. Para que quede constancia, cualquier información confidencial que el autor posee se ha excluido; si esta se incluyera, probablemente cambiaría algunas críticas, para bien o para mal.

(iii) La presente versión es un BORRADOR para el cual solicito ¡amplia retroalimentación! Nada de lo aquí escrito está fijo —especialmente todo lo relacionado con el *software* EOS.IO— y es de esperar que haya cambios.

establecer una base y una visión con el propósito de crear una infraestructura de comercio de *blockchain* que sea práctica y que rinda. Entonces, en este artículo primero, hacemos una reseña del **Contexto** del mercado actual para las DLTs. Luego, describimos una **Visión** sobre las necesidades de los usuarios finales y cómo satisfacerlas. En seguida, resumimos una **Arquitectura** que satisfaga las demandas del mercado.

Después, se hace una breve **Comparación** entre EOS y los tres sistemas más importantes de la actualidad. Finalmente, se presenta algunos **Comentarios finales**. Para obtener detalles técnicos adicionales sobre el *software* EOS.IO, se recomienda a los lectores la guía técnica de EOS.IO *"EOS.IO Technical White Paper"* (Larimer, 2017).

II. CONTEXTO

El mercado. El mercado es competitivo para todos los productos y las DLTs o *blockchains* no son la excepción. ¿Cuáles son las ofertas actuales del mercado? Bitcoin podría verse como la cadena de seguridad, pero aunque es cierto que es bastante fuerte, una cadena solo es valiosa si el negocio al que está adherida lo es también. Al reconocer este factor, Ethereum orgullosamente anunció la imparable computadora mundial Turing, una idea que quizá sea atractiva para los científicos en computación, pero no ha sido de interés para otros especialistas. R3 diseñó Corda para satisfacer las necesidades de las instituciones financieras, las cuales constituyen un mercado grande pero también uno caro y exclusivo.

Esta sección examina estos tres sistemas con base en algunas características arquitectónicas importantes que se utilizan como punto de partida y de referencia en el ámbito de las DLTs.

Consenso. Una de las características principales de una *blockchain* es que nos permite llegar a un consenso respecto de un bloque de transacciones, de tal forma que ninguna transacción entre en conflicto con otra en ningún bloque. El fundamento del consenso en las blockchain se basa en el famoso Problema de los Dos Generales. Desde hace mucho tiempo se han propuesto distintas soluciones a este problema para lograr acuerdos entre protagonistas remotos. El objetivo final es tener certeza que ambas partes puedan decir: "Yo sé que lo que tú miras es lo que yo miro"**.** Ver Figura 1.

Bitcoin estableció la prueba de trabajo *(proof of work)* o la firma Nakamoto como el sistema que permite reunir a toda una comunidad de entrada abierta en un solo libro mayor distribuido (registro compartido de transacciones), del cual todas las partes retienen una copia completa. Este mecanismo opera una lotería entre los *mineros* para determinar quién mina el bloque. Los mineros compiten en la lotería intentando resolver algoritmos SHA2 pero, debido a que este proceso requiere energía, el ganador de la lotería es premiado con una cantidad fija de bitcoin. En efecto, cualquiera puede ser un General y el que gana la lotería es el que fija el plan de batalla en ese momento. Los Generales que le siguen pueden aceptar ese plan (o bloque) o rechazarlo cuando no sea válido.

Muchos han encontrado ofensivos el concepto de un libro mayor totalmente compartido y el costo de la prueba de trabajo que este sistema conlleva (el costo del *proof-of-work* para Bitcoin es de 4% y para Ethereum es de 11% al momento que se escribió este artículo). Se ha propuesto utilizar *Libros Mayores* con Permiso (*Permissioned Ledgers)* (Swanson, 2015) no solo para bloquear a aquellos que queremos excluir de los beneficios de nuestro libro mayor, sino también para que podamos regresar a las raíces de la ciencia de la computación: un consenso eficiente que operan mediante diseños prácticos pero centralizados, en otras palabras, diseños que ya son bastante conocidos en la ciencia del almacenamiento de datos. También se ha propuesto aplicar pruebas de participación (POS, *proof of stake*), criptografías exóticas y énclaves seguros. Corda (Brown et al., 2016) estableció que el consenso lo puede decidir el usuario en puntos seleccionados dentro de un contrato de transacciones. Esto se hace por medio de servidores llamados notarios que sirven como mediadores de consenso que utiliza cualquiera de

los medios antes mencionados. La ventaja de que Corda permita que los notarios sean intercambiables es que han logrado reducir el costo operativo de su red a un nivel comparable al de la infraestructura típica de cualquier sistema operativo actual.

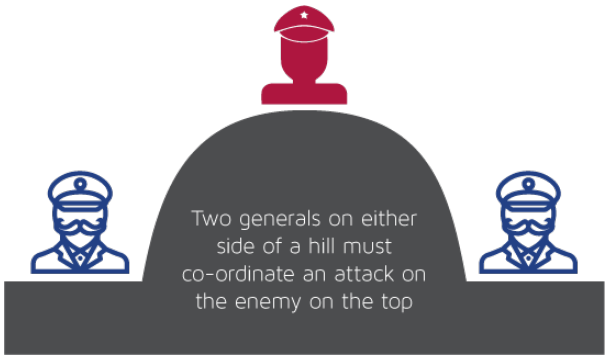

Figura 1. El problema de los Dos Generales, fundamental en la ciencia de la computación.
"Dos generales se encuentran en lados opuestos de una montaña y deben coordinar un ataque en contra de un enemigo que se encuentra en la cima de la montaña".

Valor. De forma similar, a lo largo de los años ha existido una gran variedad de mecanismos para establecer un valor fungible como el dinero en efectivo, por ejemplo. Entre 1980 y 1990, el dinero de las tarjetas inteligentes usualmente funcionaba por medio de almacenamiento de datos internos en cada tarjeta, es decir, existía un sistema interno el cual negociaba transacciones atómicas de tarjetas duales. En la misma década, el eCash de David Chaum (Chaum, 1983) popularizó la idea de una moneda que consistía en un número al azar con una firma digital ciega que podía ser transferible de un usuario a otro. Más recientemente, la contabilidad de triple entrada propuesta por Grigg (2005) establece que cada parte puede ver el mismo recibo, cada uno del cual registra una transacción de persona a persona. Por lo tanto, el balance se calcula como la suma de recibos que entran y salen.

Bitcoin usa el modelo UTXO o gasto de salidas de transacciones *(unspent transaction output)*, un diseño basado en el almacenamiento de estado. Cada registro de transacción gasta una cantidad de valores que no fueron gastados previamente y crea valores nuevos gastables para el futuro. La máquina virtual de Ethereum utiliza un mecanismo de base de datos en el que se puede emitir una criptomoneda a partir de una tabla. Este nuevo enfoque ha tenido consecuencias positivas y negativas para Ethereum: por un lado el sistema ha mejorado debido a que ahora es más flexible, pero por otro lado, se ha vuelto más vulnerable a una gran variedad de ataques.

Estos cinco mecanismos demuestran que la forma de llevar la contabilidad de un valor todavía no es una ciencia establecida.

Transición del estado. Ya que los bloques de Bitcoin funcionan como listas de UTXOs, esta *blockchain* declara como suyo el estado de la misma, refiriéndose a la naturaleza de las criptomonedas que se encuentran en un bloque dentro de una cadena en un momento determinado. La dualidad del diseño UTXO se deriva del hecho de que los clientes livianos (aquellos que utilizan la técnica de verificación simple de pago - SPV) necesitan verificar sus monedas entrantes en un libro mayor compartido. Para asegurarse que una transacción sea buena, todos aquellos clientes receptores que solamente cuentan con acceso limitado únicamente necesitan rastrear las monedas que reciben desde el último bloque hasta sus bloques de origen. El recipiendario no necesita verificar nada más que sus monedas entrantes para asegurar que tiene el control completo del valor.

Este concepto tan valioso —la blockchain es una gráfica del estado— ha sido adoptado considerablemente dentro del campo de las DLTs. Aún cuando Ethereum sustituyó el UTXO con una potente máquina virtual, Ethereum aceptó que el estado era el punto de consenso que todos los nodos deberían alcanzar. Una vez llega un nuevo bloque autenticado por una función hash, cada nodo validante calcula y se pone de acuerdo con los demás con respecto al estado de salida *(exit state)* de la blockchain, la cual está conformada por todos los contratos presentes en cada bloque nuevo.

Contratos. Bitcoin le agregó una lógica de negocios al dinero al adjuntarle 'códigos de validación' a sus transacciones, lo que creó una forma limitada de contratación que llegó a conocerse popularmente como contratos inteligentes (Szabo, 1994, 1997). La imparable computadora global Turing de Ethereum

proporcionó un sistema mucho más potente de codificación, envío de mensajes y almacenamiento de datos. Corda utilizó ambos diseños aunque en versiones más restringidas, ya que aplicó cambios de comando que volvieron más eficientes los procesos para validar y ponerse de acuerdo sobre el estado (similar a UTXO), pero limitó el acceso solamente a las partes involucradas directamente por razones de confidencialidad. Tanto Ethereum como Corda introdujeron lenguajes de alto nivel más robustos que permiten formular contratos.

Rendimiento. Bitcoin maneja un límite general de alrededor de 3 transacciones por segundo (TPS, *transaction per second*); si este límite se sobrepasa, las transacciones pueden demorarse mucho más tiempo. Ethereum parece estar llegando al límite de 15 TPS; recientemente, un evento condujo a la congestión de la red de Ethereum, lo que resultó en cobros de $2000 por transacción de alta prioridad. Existen tres factores que limitan el rendimiento en una *blockchain*: la validación de bloques previos, el procesamiento del bloque nuevo y la minería. Corda evita estas limitaciones, puesto que su consenso es por la vía de notarios seleccionables, independientes y localizados, y por lo tanto, únicamente las partes directas deben entrar en consenso. En conclusión, cada sistema tiene limitaciones físicas que afectan sus tiempos de propagación en la red.

Casos de uso. A pesar del revuelo que ha causado la tecnología de *blockchain*, existe relativamente poca evidencia concreta de casos de uso exitosos. Bitcoin emitió una sola criptomoneda, pero la explosión de los *altcoins*, el fracaso de las monedas de color *(colored coins)* y la falta de contratos inteligentes relevantes indican que Bitcoin aún tiene limitaciones significativas. Ethereum ha tratado de sobrepasar esos inconveniente, pero, hasta la fecha, no ha tenido éxito. Aunque algunos consideran que contratos como ERC-20 podrían verse como un ejemplo de éxito de Ethereum, en realidad estos contratos son en sí un tanto circulares, ya que únicamente han servido para recolectar fondos para casos de uso que aún no han sido lanzados sino que saldrán en un futuro próximo. Quizá sea sorprendente que Bitshares y Steem, los dos progenitores de EOS, son dos casos de uso 'interesantes' que han alcanzado producción y escala, siendo uno un intercambio distribuido (Bitshares) y el otro un sitio de medios sociales (Steem). Sin embargo, todavía nadie ha ofrecido cumplir con la demanda de contratos inteligentes.

Gobernanza. Para este autor, el descubrimiento más importante que aportó Bitcoin no es que podamos mediar con la criptografía, ni que el diseño es estable en términos de descentralización y entrada abierta; más bien es que la blockchain debe preservar a lo largo del tiempo estas mismas características para sobrevivir. La apertura sin restricciones no solo es clave para el modelo de consenso de la minería *hash* sobre el libro mayor distribuido, sino también es clave para la supervivencia del sistema. En el pasado, los sistemas de dinero digital fracasaron porque siempre había un centro vulnerable a ataques. Aún hoy en día, los intercambios centralizados continúan siendo atacados mediante robos, violaciones de contratos, negación de servicio, bancarrotas, allanamientos y cambios de regla forzados.

Entonces, el mundo lo podríamos dividir en dos: por un lado, tenemos los sistemas de entrada abierta totalmente descentralizados típicos de muchos sistemas de *blockchain*, y por el otro, tenemos sistemas que operan con más restricciones, tales como los de los libros mayores centralizados o con permisos. Pero todavía es incierto si estos son parte de un espectro que implica la existencia de un punto medio o si representan dos sistemas completamente opuestos e incompatibles. Por lo tanto, la bifurcación sobre el tema de las entradas abiertas presenta las siguientes preguntas: en cada uno de estos sistemas, ¿Cómo van a gobernar los usuarios? ¿Cómo serán gobernados los usuarios? ¿Cómo funcionará la gobernanza en términos de beneficios?

La estrategia general de los sistemas de entrada abierta se basa en el principio de *caveat emptor*. Estos sistemas establecen un ambiente técnico lo suficientemente robusto para que sus usuarios puedan realizar la mayor parte de lo que deseen, pero sus derechos se ven limitados por lo que la codificación ya automatiza. Etiquetado a

veces como un sistema que no requiere confianza entre las partes *(trustlessness),* este régimen marca un límite drástico entre todos los aspectos técnicos que protegen a la cadena y todo aquello que queda a discreción del usuario —y que, por lo tanto, es más peligroso. Con el paso del tiempo, han surgido algunas estrategias institucionales y propuestas de mejora. Por ejemplo, se han formado centros de poder tales como fundaciones o equipos para lidiar con algunos de los peligros para los usuarios, logrando un mayor o menor grado de éxito (Gupta, 2014). *Caveat emptor* es típico de Bitcoin y Ethereum.

Por contraste, en el enfoque de redes que operan con permisos o jardines amurallados, solo los que tienen autorización pueden entrar y actuar. En este escenario, las partes abren una cuenta, son abordados por un agente y luego pueden empezar a comerciar, asumiendo que se comportan de una manera correcta. Ya sea implícita o explícitamente, implementar reglas para asegurar el buen comportamiento es considerado como algo fuera de alcance desde el punto de vista técnico, aunque identidad típicamente juega un papel poco claro. La desventaja es que puede ser que levantar y mantener la muralla que protege al jardín cueste demasiados recursos, y además, cada año el guardia de la puerta cobra más. Este es el enfoque que utilizan los mercados altamente regulados tales como los bancos y otras entidades análogas. Corda también se rige por este método de gobernanza.

Ninguno de estos estados del blockchain es fácil para los usuarios. Por un lado, los usuarios pierden demasiado dinero en aquellos sistemas que se basan en el principio *caveat emptor*. Por otro lado, los sistemas que se rigen por 'permisos' se convierten en sistemas que discriminan, ya sea a nivel competitivo o social. Los usuarios de forma rutinaria reaccionan con escepticismo ante los dos.

III. VISIÓN

Metas finales. ¿Qué necesita nuestro usuario? En términos generales, el usuario quiere lo siguiente:

- Conocer quiénes son sus amigos, sus socios y sus clientes.
- Comunicarse con ellos:
  - A pequeña escala, hacer acuerdos entre partes iguales, y,
  - A gran escala, construir un negocio sofisticado para poder servir al mercado.
- Poder retener y dirigir su valor (pagar las cuentas, por ejemplo) como un componente necesario de los negocios.
  - Todo tiene que hacerse de forma segura.
- Poder invertir en un negocio predecible. Esto es un tema complejo, pero parece requerir de tres componentes.
  - Saber que el ecosistema está avanzando y que no está en riesgo de fracasar.
  - Pagar por adelantado con la seguridad de que obtendrá un retorno razonable en el futuro.
  - Una estrategia confiable que le permita resolver sus problemas de forma rápida, económica y sin complicaciones indebidas. Esto es necesario, ya que el usuario sabe que las cosas – contratos, bienes, transacciones, intentos – pueden salir mal en distintos ámbitos y con distintas partes (sus amigos, su negocio y sus bienes, etc).

ACLARACIÓN (de nuestra propia arrogancia): asumimos que los deseos y las necesidades de nuestro usuario son sinónimos. Mejor dicho, estamos interpretando de forma empresarial lo que nosotros creemos que son sus necesidades y lo que el cliente querrá una vez conozca *y* entienda el sistema.

Idea principal. Ha quedado muy claro que, por una razón u otra, la promesa de contrataciones y dinero universales entre una parte y la otra todavía no es una realidad en la Internet. Bitcoin es demasiado inseguro y sus contratos inteligentes son opacos. Ethereum es demasiado intimidante, difícil y útil solo para los técnicos. Corda es 'el gran corporativo'. Aunque otros sistemas también poseen sus

respectivas debilidades, todos se asemejan en el sentido que son restringidos a los codificadores élites, y por lo tanto, cada uno de ellos opina y tiene un punto de vista distinto.

Se necesita una opción de negocio inteligente para la persona común y corriente. Es imprescindible que contemos con una aplicación distribuida que podamos usar de forma cotidiana en una *blockchain* global que cumpla con los siguientes requisitos: operar bajo el atesorado concepto de entrada abierta que nos regaló Bitcoin, tener suficiente capacidad para construir un negocio de gran escala, estar lo suficientemente conectado para reunir a las personas, y ser lo suficientemente seguro y confiable como para que *Gordon Gecko* (la persona común) de Wall Street pueda hacer negocios con *Mama Biashara* (otra persona común) del África. Sin drama, sin temor, sin perderse de nada.

El usuario objetivo. Nuestra visión implica desarrollar una sola *blockchain* de contratación global que logre manejar una larga lista de negocios que necesiten comercializar contratos de ventaja mutua en un ambiente seguro y confiable.

En términos más prácticos, aunque haya muchas cosas de valor en la Internet, nos enfocamos en lo que se media en la web y, por ahora, dejamos a un lado los servicios de medios móviles y sus aplicaciones asociadas. ¿Qué quieren los desarrolladores de aplicaciones web? Ya que asumimos que el usuario objetivo es el desarrollador de aplicaciones web, vamos a trabajar a partir de su perspectiva.

Características principales. Nuestro diseño predice que la *blockchain* de EOS podrá manejar miles de transacciones por segundo (TPS) para contratos inteligentes de negocios y que estos contratos podrán ser desarrollados en lenguajes seguros y fáciles de usar. Las principales características de nuestro diseño incluyen:

- Sistema de alto rendimiento basado en mensajería/eventos.
- Prueba de participación delegada (DPOS, *delegated proof of stake)*
- Contratos como acuerdos de negocio y de intención – en esencia, un sistema de mensajería
- Versatilidad de uso para el usuario, el redactor del contrato, el desarrollador y el empresario.
- Gobernanza para el mantenimiento del negocio y de la cadena.

La siguiente sección explora más a fondo estos temas.

IV. LA ARQUITECTURA

Filosofía. En buena medida, el enfoque que se utilizó para diseñar el *software* de soporte de EOS fue extender la experiencia de las *blockchains* de gran escala y alto rendimiento de Bitshares y Steem con el fin de apoyar los negocios de usuarios finales. La mayoría de los elementos de nuestro diseño han dado resultados adecuados en mayor o menor grado en el pasado; esta arquitectura las re-ensambla con otro fin: construir aplicaciones distribuidas (Dapps).

Este inciso describe algunas diferencias arquitectónicas que el *software* de soporte de EOS propone en contrapartida a prácticas previas. Para los detalles más técnicos, los lectores pueden referirse al *EOS.IO Technical White Paper* (Larimer, 2017).

El mensaje es el medio. En lugar de basarse en el popular principio del consenso sobre estado, el diseño de *software* de EOS utiliza un concepto relativamente desconocido: consenso sobre mensajes (o eventos) (Grigg, 2017-1) Este enfoque vincula la estrategia de determinar el origen con base en mensajes o eventos (event sourcing pattern) (Fowler, 2005) con una *blockchain* conformada por eventos en lugar de estado.

En la ciencia de la computación, las máquinas que determinan estados están construidas como máquinas de código, de estado (memoria) y de eventos, tanto entrantes como salientes. Cada vez que algo ocurre que cause un cambio, una máquina práctica guarda los intermediarios de la memoria y, al reiniciarse, la máquina recupera su memoria al interpretar dichos intermediarios. Cuando se construye una máquina de estado práctica, podemos decidir si

almacenamos eventos o el estado, una decisión que depende principalmente en qué estamos tratando de optimizar.

En la Figura 2, ¿debemos guardar los mensajes rojos o el estado azul? Una máquina que almacena el estado probablemente se use más en un contexto donde la prioridad es determinar el estado de un momento específico, por ejemplo, en las bases de datos. Una máquina que guarda mensajes probablemente sea más útil cuando sea más importante preguntarnos cómo llegamos al estado en el que estamos en este momento, por ejemplo en los protocolos o en los libros de registro con poder legal tales como contabilidad de triple entrada (Grigg, 2005). Aunque ambos procesos son útiles, estos difieren en lo siguiente: el reinicio de una máquina es más rápido si se guarda el estado; el proceso de enviar de un punto a otro es más rápido cuando se almacenan mensajes.

Debido a que los usuarios priorizan el rendimiento, nuestro diseño está basado en almacenar mensajes. Reiniciar una máquina de mensajes o de eventos equivale a recuperar la información comenzando de cero; por lo tanto, es increíblemente lento. El optimizar el reinicio implica guardar puntos de verificación – en otras palabras, podemos derivar el estado. Sin embargo, y aquí hay un punto crucial, al guardar ese estado, un participante queda vinculado por los mensajes guardados, no por el estado. Por lo tanto, podemos optimizar muchísimo y hasta recalcular los puntos de verificación, si fuera necesario. La forma precisa de cómo se hace esta optimización es un tema demasiado complejo para fines de esta introducción. Sin embargo, se podría predecir que la combinación de ambas técnicas podrían, en teoría, ayudar a que una *blockchain* pase de realizar las típicas 3 TPS a efectuar la cantidad de 3 millones TPS.

Consenso. Para llegar a los consensos sobre mensajes, la arquitectura EOS.IO usa la prueba de participación delegada (DPOS, *delegated proof of stake*), una estructura de gobernanza de dos niveles puesta a prueba en Steem y Bitshares (Larimer, 2014). En el primer nivel, los productores de bloque son elegidos para participar en una ronda de 21 productores. Cada uno de los 21 productores de esa ronda

Figura 2. Modelo de una máquina expendedora de gaseosas en términos de una máquina de estado.
Las flechas rojas de la izquierda representan la moneda y el botón (los mensajes). El botón azul representa los estados. La flecha derecha representa la bebida. El Estado 1 indica que si el mensaje "Moneda" es detectado, entonces se debe proceder al Estado 2. Si el mensaje "Botón" es detectado, entonces se emite el mensaje "Bebida".

recibe un bloque y es premiado por validar los mensajes entrantes y por producir bloques de mensajes. El bloque que libera un productor es validado por el que sigue y el que sigue y así, contínuamente. Si dicho bloque no es validado, no se construye sobre él. El proceso continúa aplicando mecánicas de la cadena más larga similares a las que utiliza Bitcoin y, seguidamente, los productores convergen en una cadena más larga. Un bloque aceptado por un quórum de productores se declara inmutable y la cadena de bloques inmutable se convierte, de hecho, en punto de verificación.

Como con la prueba de trabajo, los productores pueden censurar (ignorar) los mensajes o pueden adelantarse e introducir uno propio siempre y cuando tengan conocimiento superior acerca del futuro. Para aplicar un sistema de gobernanza mesurada sobre las malas acciones de los productores, cada ronda de productores es elegida contínuamente por la comunidad usando el principio de prueba de participación. Ya que la elección mediada por *blockchain* del segundo nivel del DPOS es sobre los productores y no sobre los bloques, es imposible argumentar que el sistema sufre de la debilidad de "nada que arriesgar".

En efecto, se escoge un grupo de Generales para una campaña y cada uno tiene un turno. Después de la campaña, la comunidad civil opina si se reemplaza o no a los Generales que no rindieron adecuadamente.

El DPOS evita el impuesto de minería, lo que implica que se libera un valor sustancial que

regresa a las partes interesadas del proyecto. El valor de los premios por la validación de mensajes y producción de bloques inicialmente sería captado enteramente por los productores. Sin embargo, puesto que estos son electos por la comunidad, tienen el incentivo de compartir los premios mediante un esquema que acuerden los productores entre ellos y que promueven ante la comunidad.

De acuerdo a la Constitución, el premio a largo plazo por producir bloques se debería limitar, por ejemplo, al 5% anual (Larimer, 2017-2). En la práctica, sugerimos que se retorne la mayor parte del valor a la comunidad para el bien común – mejoras en el *software*, resolución de conflictos y cosas similares. Manteniendo el espíritu de 'probar nuestro propio producto', el diseño contempla que la comunidad vote sobre un conjunto de contratos de entrada abierta que sirvan como los "cimientos" que beneficiarán a toda la comunidad. Este mecanismo se le conoce como Contratos de Beneficio Comunitario, el cual resalta la importancia del DPOS para facilitar la gobernanza directa de la comunidad sobre las cadenas.

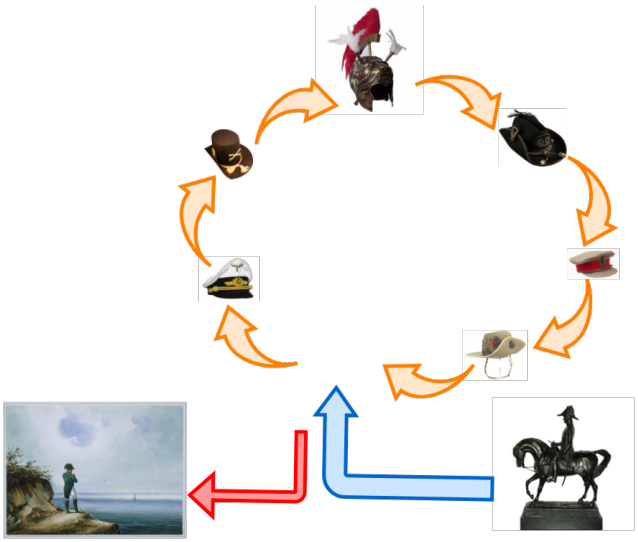

Figura 3. Delegar permite que se pueda reemplazar a cualquier General que haya tenido una mala campaña

El contrato. El diseño o arquitectura se acerca más a la naturaleza de la contratación porque trata los contratos como una expresión dinámica de negociación, compromiso y eventos, en lugar de interpretarlos de forma más estática como 'las cuatro esquinas de una página' o el código de rendimiento dentro de una máquina. Proponemos que los mensajes constituyen en esencia los elementos de un contrato, puesto que captan mejor todas las fases de una contratación exitosa: la negociación, la intención, el rendimiento y el incumplimiento de las obligaciones son todos eventos que se captan mejor como mensajes que como estados.

Un usuario escribe un contrato como una construcción virtual de gestores de manejadores de mensajes. Un usuario puede convertir su cuenta en un agente contratante si incorpora manejadores de mensajes y si usa el almacenamiento de datos inherente en su cuenta para retener la posición interna de sus contratos. Varios manejadores de mensajes trabajando juntos pueden mediar un flujo de mensajes para poder llevar a cabo un contrato completo o un acuerdo legalmente confiable de principio a fin.

Desde la perspectiva de un contrato, la llegada, aceptación y procesamiento de un mensaje implican una abstracción más sencilla que si se hiciera con un estado. Consideremos, por ejemplo, un libro de procesamiento de pedidos en un mercado de valores: el libro registra ofertas de compra y de venta. Cuando llega el momento adecuado, debe calcular un precio de punto de equilibrio, para luego emitir pedidos para ambos lados.

Un libro de pedidos dentro de un sistema basado en mensajes se compromete a un conjunto de mensajes entrantes y un conjunto de mensajes salientes, lo cual representa una tarea relativamente fácil de rastrear. En contraste, en un sistema basado completamente en el estado, todos los negociadores tienen que negociar entre sí el estado que consideren aceptable — incluyendo cantidades, precios y una gran cantidad de partes involucradas —, antes de someter a consideración un estado final a la *blockchain*. Esto implica que los negociadores podrían entrever la solución, antes de llegar al acuerdo general, y se abriría la posibilidad de hacer trampa. En la práctica, la única forma que sabemos puede resolver este problema es con agentes y mensajería. Un agente activo recibe

mensajes ejecutados (committed messages), decide el resultado, y envía mensajes que se ejecutan ante ese resultado.

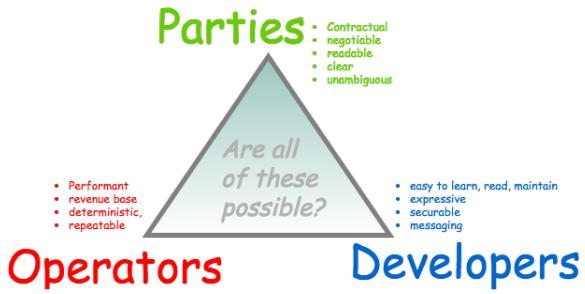

Figura 4. Tensión entre las tres partes involucradas de la *blockchain.*
Partes (contractual, negociable, fácil de entender, legible, sin ambigüedades)
Operadores (trabajo con base en ganancias, deterministas, repetibles)
Desarrolladores (fácil aprender, leer y mantener, capaz de expresarse, asegurable, mensajería)

Uso. El usuario directo de una *blockchain* es el desarrollador que crea aplicaciones web para sus usuarios finales. Para lograr apoyar a un usuario final, el *software* debe apoyar, antes que nada, al desarrollador, y se debe hacer de manera que ayude al desarrollador a apoyar a sus usuarios. Las características que determinan la calidad del apoyo que el *software* le brinda al desarrollador incluyen: (a) las herramientas, (b) el lenguaje y (c) el ambiente.

Por lo general, el desarrollador EOS.IO tendrá a su disposición un kit de herramientas en línea que le proporcionará un marco de servicio completo, sobre el cual podrá construir aplicaciones distribuidas sobre una *blockchain*. Este *toolkit* contará con elementos como cuentas, nombramientos, permisos, recuperación, almacenamiento de base de datos, horarios, autenticación y la comunicación asíncrona inter-app. Una de las metas de esta arquitectura es proveer al constructor de aplicaciones web un sistema operativo completamente equipado. Nuestro sistema se enfoca en aplicaciones web, ya que la mayoría de los usuarios utilizan con mayor frecuencia este tipo de aplicaciones.

Lenguaje. Dentro de la industria de las Dapps, el lenguaje para programar contratos inteligentes debe ser considerado prioridad debido a que es uno de los elementos que mayor valor le agrega al sistema. Casi todas las demás características arquitectónicas en el *software* EOS.IO ya fueron probadas en Bitshares y Steem, pero programar contratos inteligentes en una *blockchain* es prácticamente territorio desconocido.

Por lo tanto, tenemos que analizar de forma meticulosa las características que requieren un buen lenguaje de programación para este diseño. Desde el punto de vista de seleccionar la tecnología adecuada para automatizar contratos inteligentes, los tres actores claves que son críticos para el éxito son: las partes, los desarrolladores y los operadores.

- Las partes necesitan un contrato que sea, primeramente, un contrato verdadero. Las partes también necesitan que el contrato sea negociable, legible, claro y sin ambigüedades – necesitan que su intención humana se capte fielmente. De preferencia, los contratos también deben contar con opciones que permitan resolver disputas o desacuerdos y que aseguren el cumplimiento de las obligaciones de todas las partes involucradas.
- El desarrollador necesita que tanto el lenguaje como el sistema sean fáciles de aprender y que se pueda escribir en ellos. Además, el lenguaje debe ser lo suficientemente versátil para poder redactar conceptos complejos y ser seguro; estas son características que a menudo no tienen relación con la semántica o la intención contractual.
- Los operadores de la *blockchain* – productores de bloques y de negocios de aplicaciones de nodo completo *(full-node)*– necesitan que el contrato sea escalable y que les permita obtener ganancias, intereses que tienen poco que ver con la intención humana de las partes o el potencial de expresión del desarrollador.

Si tomáramos en consideración las necesidades de las partes, primero necesitaríamos combinar la prosa legal (en texto sin formato) y el lenguaje de codificación, e incluir algunos parámetros que nos permitieran "impulsar el acuerdo" y reutilizar la misma prosa

y el mismo código para muchos contratos (Grigg 2015). Muchos esfuerzos de investigación han tratado de fusionar los dos elementos de la contratación inteligente — código y prosa legal—, ya sea como parámetros de orden superior o como un lenguaje legalmente expresivo de dominio específico (Clack1 et al., 2016, ver su Figura 5). Sin embargo, ninguno, hasta el momento, ha descubierto cómo hacerlo. Esta es un área de investigación abierta que ha resultado en varias posibilidades de diseños, pero ninguno ha sido aprobado en la comunidad (Clack2 et al., 2016).

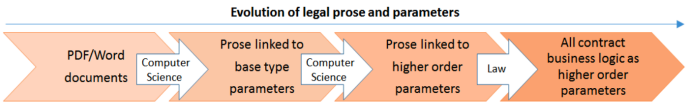

Figure 5: Parameters may become more sophisticated in the future, evolving from just simple base type parameters to also include more complex higher-order parameters. In the far future, if the encoding of business logic used in the parameters becomes acceptable to lawyers and admissible in court, then it could potentially replace the corresponding legal prose.

Figura 5. Evolución de conceptos de automatización de códigos y prosa contractual (Figura 5 de Clackl)

Siguiendo esta línea, nuestra primera tentación fue inclinarnos hacia lo que busca el desarrollador: un lenguaje interpretado de código fuente *(source-interpreting scripting language)* basado en Wren y adecuado para administrar el diseño de un manejador de mensajes contractuales. A continuación, un extracto del código (Larimer 2017-1):

```
apply:
     // assuming all prior steps pass,
     // perform the state transition
     // that updates balances and/or
     // creates a new account for receiver
     var from = Balance[message.from]
     var to   = Balance.find( action.to )
     from.bal = from.bal - action.amount
     to.bal   = to.bal   + action.amount
```

Este híbrido de Wren es sencillo de aprender, leer y razonar, lo cual lo hace ideal para la contratación automatizada. Sin embargo, en nuestras pruebas resultó muy lento: una prueba de transacciones triviales topó en 1,000 TPS, lo que nos causaría conflicto debido a que no lograríamos satisfacer las necesidades de los operadores, los productores y los negocios de aplicaciones.

Ya que queremos que nuestro sistema opere 100 veces más rápido, el equipo decidió utilizar WebAssembly (WASM, en inglés), un lenguaje intermedio nuevo diseñado para hacer el trabajo que Javascript hace actualmente en los navegadores web. La primera prueba no-optimizada de WASM dentro del marco EOS rindió unas 50,000 TPS para un contrato de divisas.

Sin embargo, WASM transfiere el reto de los operadores a las partes – ahora hay 3 perspectivas tangibles sobre cualquier contrato: la prosa legal, el código fuente inicialmente en C, y el código intermedio en WASM.

Por lo tanto, es razonable preguntar - ¿qué es o dónde está el contrato que las partes acordaron? Quisiera responder de manera frontal a esa pregunta. Durante las dos décadas, más o menos, en que he visto que se emiten contratos en la red —ya sea Ricardiano o de cualquier otro modelo— y las cientos de emisiones que han surgido de los mismos, aún no he visto una disputa, ni siquiera una confusión en la que lo que decía el contrato o lo que significaba era la clave del desacuerdo. Aún en el incidente de *The DAO,* —esa desafortunada lección de $150 millones sobre cómo no emitir un contrato— la causa probable del *hack* fue por cuestiones de seguridad. Aunque habían varias desacuerdos e interpretaciones con respecto *al significado contractual del hack*, la única respuesta para solventar el problema fue cambiar arbitrariamente lo que había que cambiar para recuperar el dinero. No hubo ni siquiera un mínimo intento para resolver la disputa sobre la interpretación de los hechos, el significado y los derechos. Todavía está abierta la pregunta sobre qué proporción de las disputas en las cortes se basa por interpretaciones ambiguas de significados o por confusiones, y qué porcentaje son simplemente juegos de poder e intimidación; pero no soy optimista.

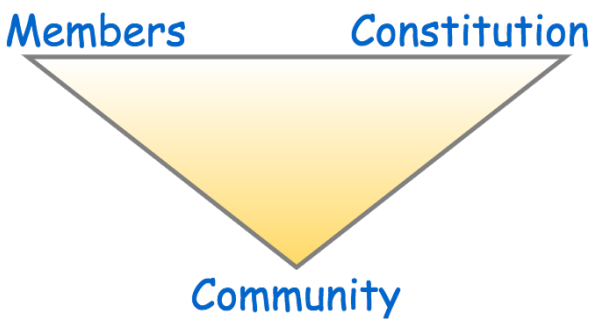

Figura 6. Los miembros forman una Comunidad regida por una Constitución.

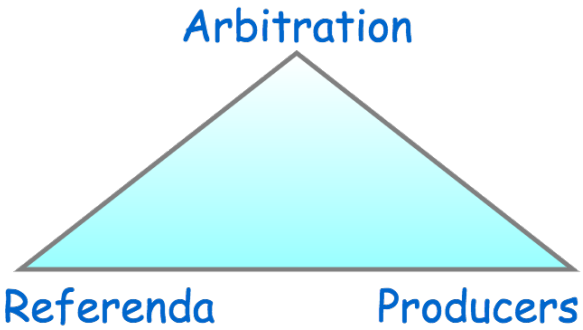

Figura 7. La Comunidad elige gobernadores que gestionan responsabilidades.

Ante *El DAO* y otras experiencias, sugiero que la regla de un único contrato (Grigg 2004) parece ser dogmática y exageradamente constrictiva. En lugar de eso, por lo menos para la parte no-regulada de las DLTs, hay oportunidad de liberar los componentes del contrato para lograr un mejor rendimiento, aún si se abre la pequeña posibilidad de una discordancia. Mientras tanto, debemos enfocarnos en la gobernanza y en lograr que la resolución de desacuerdos esté disponible y sea cómoda para las partes.

En el momento en que se redactó este documento, todavía se está trabajando en las opciones de lenguajes que estarán disponibles para los desarrolladores de contratos. Ya sea con WASM o Wren u otro, seguiremos teniendo que estructurar el lenguaje para que rinda y se pueda usar. Cada manejador de mensajes tendrá que identificar las secciones en las que quiera utilizar código estático, solo lectura (*read-only)*, y de lectura y escritura (*read-write)*, cada uno de los cuales tiene diferente potencial de optimización. Para eliminar problemas de reentrancia, los mensajes salientes se almacenarán hasta completarse o serán descartados si el sistema tiene una falla. Nuestra intención es agregar una estructura de tabla similar a SQL (lenguaje de consulta estructurada) para que aquellos que están familiarizados con los sistemas convencionales de bases de datos puedan adoptar nuestra estructura fácilmente. La criptografía será externa, casi enteramente invisible.

Así como en los demás espacios de DLT, la competencia continúa internamente. Wren ofrece un espacio pequeño y compacto. WASM a penas fue estandarizado hace poco tiempo. Las primeras herramientas de WASM fueron diseñadas para lenguajes C y C++, pero a pesar que estos son bastante populares, implican un mayor costo al momento de escribir códigos, en comparación con lenguajes de mayor nivel y de generaciones avanzadas, tales como Wren. Estos retos probablemente no sean imposibles de solucionar a largo plazo, ya que el proyecto WASM está diseñado para trabajar con la mayoría de los lenguajes, y la mayor parte del código de cualquier Dapp está fuera del alcance de los manejadores, en los sitios de la red. La idea de un sistema versátil que acepta muchos lenguajes es muy atractiva. A pesar de que esta es una ventaja con la que podría contar Corda gracias a la implementación de JVM, ni Bitcoin ni Ethereum podrían tener esta flexibilidad a menos que adopten un enfoque más holístico con respecto al ciclo de desarrollador.

En conclusión, seleccionar lenguajes y cajas de herramientas aptas para el desarrollador va mucho más allá que simplemente escoger las mejores opciones que puedan ser codificables. Nos gustaría un lenguaje de programación que sea fácil de leer y entender, que pueda expresarse en términos contractuales completos, que sea asegurable y que sea escalable. Sin embargo, con el estado actual de este "arte", no tenemos más opción que hacer compromisos.

Gobernanza. Consideremos ahora el ambiente. La realidad es que muchas cosas salen mal cuando se automatizan los contratos, lo que preocupa a todos. Esperamos reducir tanto la frecuencia como los costos de esos errores, pero estos nunca podrán ser eliminados totalmente. Nuestra solución está en incorporar métodos remediales para cuando ocurran.

Una *blockchain* basada en *software* EOS.IO asume que todos los que usan dicha *blockchain* son miembros que trabajan bajo una misma Constitución (Larimer, 2017-2) (Grigg, 2017-3) y, al comprometerse todos a la misma, todos los miembros forman una comunidad sujeta a la Constitución.

La Constitución establece algunas reglas básicas para el beneficio de la comunidad. La Constitución le otorga poder a tres brazos de gobernanza: los árbitros para resolver disputas, productores de bloques para escoger bloques, y los referendos para darle voz a la comunidad. Distribuidos en un triángulo entrelazado de gobernanza, estos tres brazos se apoyan y equilibran entre sí. La comunidad usa los referendos para elegir a los productores y a los árbitros, así como los cambios al código y la Constitución. Los árbitros pueden emitir resoluciones legales para resolver desacuerdos, así como para aprobar cambios extraordinarios, como en el caso de las bifurcaciones duras *(hard forks)*. Los productores de bloques tienen la libertad técnica de censurar malas transacciones o de introducir algunas remediales – pero están conscientes de la reacción de la comunidad. Los árbitros publican resoluciones que los productores pueden hacer valer, o los usuarios pueden buscar coacción externa.

Este arreglo de contrapesos asegura que ninguna parte o grupo tenga poder total. Hasta los mismos fundadores o desarrolladores tienen poder limitado que pueda afectar los derechos de los miembros de la comunidad. Contamos con instrucciones definidas para los *hard forks* y otras actualizaciones. Las disputas individuales se canalizan hacia un lugar específico que permite resolver el problema y regresar a las operaciones rápidamente y sin consecuencias mayores. Otro beneficio es que casi toda la gobernanza antes mencionada se puede administrar de forma transparente. Es decir, se programa que los gestores de contratos acepten y controlen las disputas, manejen los referendos y asuntos semejantes.

Para que estas instituciones funcionen, los usuarios tienen que aceptar la Constitución que le da el poder a los productores de escoger los bloques y reserva toda disputa para el foro de arbitraje. Así mismo, la Constitución crea los derechos legales expresados en la *blockchain* al declarar que cada miembro recibe esos derechos que están apropiadamente revisados y, en respuesta, cada miembro apoya los derechos revisados de otros. Este sistema de «tus derechos por los derechos de otros» se convierte en la piedra angular de la comunidad en el sentido de que la comunidad es definida tanto por el uso de la plataforma como por la aceptación de la Constitución.

Así es como hemos preservado el principio de entrada abierta, a pesar de que la comunidad se auto-gobierna de forma interna. Aún cuando un usuario lleva a cabo transacciones, todas las transacciones, desde la primera entrada hasta la más reciente, se refieren a la Constitución por medio de *hashing,* como se hace en un contrato Ricardiano (Grigg, 2004). Como mecanismo explícito de gobernanza, la Constitución crea un modelo más parecido a un campo cercado que un jardín amurallado, y el guardia de la puerta se transforma en una transacción o una señal que se encuentra en todos los puntos.

V. COMPARACIONES

Bitcoin. Al ser la plataforma que lanzó la primera y más exitosa criptomoneda, Bitcoin es una base de referencia. Sin embargo, debido a que es pionera en su campo, sus fallas sobresalen tanto como sus éxitos: el modelo de verificación UTXO implica que los negocios inteligentes se tienen que mediar por medio de códigos externos. Aunque el estado es completamente inmutable en esta *blockchain*, lo más difícil es el proceso de negociación y esta tarea únicamente se puede hacer por medio de aplicaciones. Bitcoin no ofrece un marco apropiado para el manejo de bienes, en particular porque cada transacción requiere de BTC. Por lo tanto, es una afrenta a la apócrifa Ley de Gresham en contra de la mezcla de bienes, *el dinero bueno expulsa el malo.* La falta de un sistema de gobernanza bien pensado produce mucha dificultad para realizar actualizaciones y la comunidad entra en guerra consigo misma. Por ejemplo, el límite artificial de 3 TPS que mata su potencial de

escalabilidad se debe a la ausencia de gobernanza.

Ethereum. Para corregir las debilidades de Bitcoin, Ethereum establece una capacidad de un sistema Turing completo virtual en una computadora de alcance mundial. Pero posee varios defectos graves. Primero, el obligar al sistema a encontrar consenso sobre el estado en miles de ejecuciones de programas resulta bastante restrictivo, ya que produce una congestión de recursos alrededor de los 15 TPS. Segundo, la decisión de manejar de forma independiente los lenguajes, VMs, kits de herramientas y otros elementos similares entorpece el trabajo de los desarrolladores. Tercero, padece de la *adhocracia* de la Fundación que ha emergido, a pesar de que todas las partes involucradas se niegan a reconocer la necesidad de una gobernanza. Como una propuesta emergente de negocios, uno de los mayores éxitos de Ethereum ha sido servir como una plataforma donde se recauda fondos para proyectos dirigidos en su mayoría para terminar con Ethereum o para competir contra ella. Hay muy pocos casos de uso innovadores que han dejado su marca, lo cual sugiere que falta más trabajo por hacer antes de que el concepto Ethereum de contratos inteligentes rinda sus frutos.

Corda. El principal factor distintivo de Corda es que no es una *blockchain* sino un marco para el flujo de trabajo entre una parte y otra. En lugar de subir contratos y acciones a una *blockchain*, las partes intercambian mensajes y llegan a consenso vía los notarios. Logra proteger la confidencialidad de las partes, demuestra un alto rendimiento no constreñido por la coordinación en cadena y le otorga a las partes la habilidad para controlar los contratos conforme van teniendo éxito o fracaso. Sin embargo, el flujo de trabajo funciona mejor cuando hay pocas partes y no cuando existe una gran cantidad de involucrados. Por lo tanto, este sistema es más débil en cuanto a la emisión de bienes, especialmente de dinero en efectivo y en el intercambio de efectivo denominado. Otra debilidad es que el enfoque del jardín amurallado de Corda para los negocios regulatorios le impide ser un mercado masivo atractivo para los pequeños jugadores.

VI. CONCLUSIÓN

La experiencia del usuario. Los usuarios directos de una *blockchain* como EOS son los empresarios y desarrolladores que escriben contratos para implementar aplicaciones distribuidas o Dapps. Sus usuarios son los clientes rutinarios en menudeo, finanzas, logística y medios de comunicación. Estos clientes no necesitan saber qué es una *blockchain*. Por lo tanto, la meta está en darle a los desarrolladores una plataforma que permita que se construya una lógica extensa de negocios, pero donde los mecanismos de la comunicación estén fuera del alcance de los usuarios finales.

Al desarrollador Dapp se le da una plataforma completamente equipada para manejar cuentas, permisos y mensajes, en donde se puede diseñar un sistema complejo. La interface del usuario combina lo que es familiar para los usuarios – un *webkit* para construir sitios web— y, por supuesto, acceso a la *blockchain*. Este enfoque implica que estamos ofreciendo «un sistema operativo para *blockchain*».

El hecho de que hay una *blockchain* se le puede ocultar al usuario, tal y como se hace en Steem, la cual es vista simplementemente como una plataforma de blogging que resulta que es distribuida en una *blockchain*.

Casos de uso. La intención de la *blockchain* de EOS es que se use para mensajería de alto rendimiento con lógica de negocios. Se espera que los casos de uso más populares de esta *blockchain* incluyan sistemas de cadenas de abastecimiento, administración de recursos, mensajería de usuario tal como los medios sociales, emisión de bienes y comercio, contabilidad para remesas y juegos en línea.

Un caso típico podría ser Uber. El compartir transporte está basado en fijar estándares de comportamiento para el conductor y el pasajero. Si los conductores y pasajeros fueran parte de la misma comunidad, habría un beneficio inmediato – la base de las obligaciones y de los estándares de comportamiento estarían cubiertos bajo la constitución comunitaria y la

resolución de conflictos, y sus contratos podrían ser bilaterales en lugar de ser intermediados. Así, se minimizaría cualquier dificultad regulatoria.

Entonces, como los contratos pueden ser bilaterales, el flujo del negocio se podría dividir en lo siguiente: seguirle la pista a los pasajeros en el mercado, seguirle la pista a los carros disponibles, encontrar un *match*, negociar un contrato, rendimiento, conciliación, precios y seguimiento social – todos podrían construirse como Dapps separadas que interactúan entre sí.

Comunidad. Para apoyar a los negocios, necesitamos darles las herramientas adecuadas para resolver problemas. Y para graduar la resolución de los problemas, la propia comunidad tiene que ser capaz de hacerlo. Esto significa que tiene que estar incorporada en la arquitectura. Para lograr que la comunidad avance, debemos preservar el principio de entrada abierta pero, una vez hay nuevas incorporaciones a la comunidad, debemos brindarle las herramientas a los usuarios para asegurar una buena gobernanza. Los usuarios usualmente quieren poder definir sus riesgos y obligaciones ante sus contrapartes.

Cuando están comprometidos como comunidad bajo una Constitución, los usuarios sabrán que los derechos, las responsabilidades contractuales y las obligaciones de sus contrapartes están por lo menos ajustadas a un estándar básico, como se expresa en una Constitución y como se hace valer en la resolución de conflictos. Adicionalmente, usar nombres confiables y trabajar en una red de confianza pueden reducir la anonimidad de la Internet y darle a la gente un sentido de pertenencia a algo importante.

RECONOCIMIENTOS

Este documento recibió retroalimentación de parte de Brendan Blumer, Arthur Doohan, Dan Larimer, Wendy Lee, Aaron Leibling, Konstantinos Sgantos, Joseph Vaughn Perling, Kokuei Yuan.

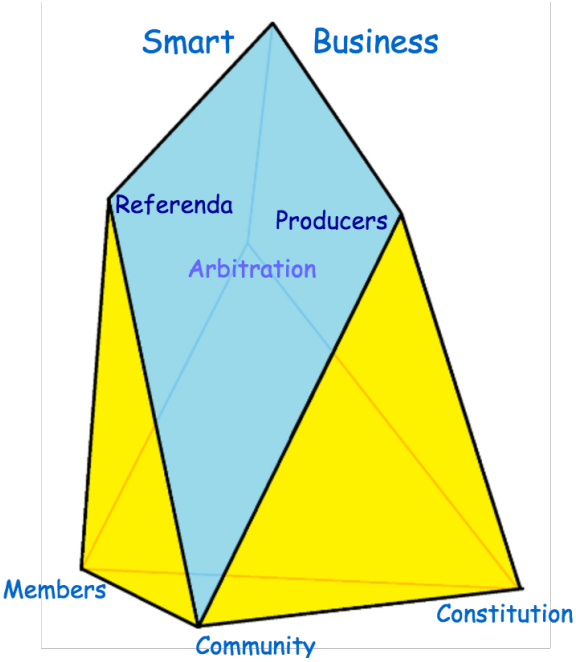

Figura 8. Negocios inteligentes.

REFERENCIAS

1) Richard Brown, James Carlyle, Ian Grigg, Mike Hearn, “Corda: an Introduction” 2016
2) David Chaum, “Blind Signatures for Untraceable Payments”, 1982 UC Santa Barbara http://blog.koehntopp.de/uploads/Chaum.BlindSigForPayment.1982.PDF
3) Christopher D. Clack (1), Vikram A. Bakshi, Lee Braine “Smart Contract Templates: foundations, design landscape and research directions”, 2016
4) Christopher D. Clack (2), Vikram A. Bakshi, Lee Braine “Smart Contract Templates: essential requirements and design options”, 2016
5) Martin Fowler, “Event Sourcing”, 2005 https://martinfowler.com/eaaDev/EventSourcing.html
6) Ian Grigg, “The Ricardian Contract,” 2004
7) Ian Grigg, “Triple Entry Accounting,” 2005
8) Ian Grigg, “The Sum of All Chains - Let’s Converge,” 2015 Ian Grigg, blog post “The Message is the Medium,” 2017-1
9) Ian Grigg, blog post “Seeking Consensus on Consensus,” 2017-2
10) Ian Grigg, blog post “A Principled Approach to Blockchain Governance” 2017-3
11) Vinay Gupta, interview “Bitcoin Cannot be divorced from pre-existing political theory,” 2014
12) Daniel Larimer, “Delegated Proof-of-Stake (DPOS)” 2014.
13) Daniel Larimer, Charles Hoskinson, Stan Larimer, “A Peer-to-Peer Polymorphic Digital Asset Exchange” 2014.

14) Dan Larimer, “EOS.IO Technical White Paper” block.one 2017 https://github.com/EOSIO/Documentation/blob/master/TechnicalWhitePaper.md
15) Dan Larimer, block post “Implementing a Hypothetical Currency Application on EOS,” 2017-1 https://steemit.com/eos/@eosio/implementing-a-hypothetical-currency- application-on-eos
16) Dan Larimer, blog post “What could a blockchain Constitution look like?” 2017-2
17) Satoshi Nakamoto, “Bitcoin: A Peer-to-Peer Electronic Cash System ” 2008
18) Tim Swanson, “Consensus-as-a-Service” 2015 http://www.ofnumbers.com/wp-content/uploads/2015/04/Permissioned-distributed-ledgers.pdf
19) Nick Szabo, “Smart Contracts”, 1994
20) Nick Szabo, “Formalizing and Securing Relationships on Public Networks”, 1997
21) Gavin Woods, “Ethereum: A Secure Decentralised Generalised Transaction Ledger”, 2014