BlockVOTE: una arquitectura de un sistema de votación electrónica basado en blockchain

1s tChinnapong Angsuchotmetee
Departamento de Ciencias de la Computación
Facultad de Ciencias
Universidad Príncipe de Songkla
Songkhla, Tailandia
chinnapong.a@psu.ac.th

2Dakota del NortePisal Setthawong
Departamento de Sistemas de Información Gerencial
Universidad de la Asunción
Samut Prakarn, Tailandia
pisalstt@msme.au.edu

3rdSapjarern Udomviriyalanon
Departamento de Ciencias de la Computación
Facultad de Ciencias
Universidad Príncipe de Songkla
Songkhla, Tailandia
5910210111@psu.ac.th

Resumen—Los sistemas de votación electrónica brindan muchas ventajas sobre los sistemas de votación tradicionales basados en boletas, principalmente sobre la precisión y la velocidad del proceso de conteo de la votación. Sin embargo, los sistemas de votación electrónica sufren muchos problemas técnicos y de seguridad que han limitado su implementación en escenarios de votación, como la votación de empresas y las elecciones políticas. Los sistemas centralizados de votación electrónica no son seguros por naturaleza y existen muchas vías de ciberataques que podrían alterar el resultado de la votación. El sistema de votación electrónica debe ser altamente seguro, garantizado a prueba de manipulaciones, y la votación debe ser digna de confianza. En este estudio, proponemos BlockVOTE, un sistema de votación electrónica basado en Blockchain. Nuestra propuesta utiliza Blockchain para garantizar que el proceso de votación se mantenga seguro y confiable a través del mecanismo de manejo de consenso de Blockchain. El diseño de la arquitectura y la sugerencia de implementación se proporcionan en este estudio. La implementación de la propuesta fue desarrollada y probada a través de la experimentación. El resultado del experimento y la discusión sobre la posibilidad de adoptar nuestra propuesta en una elección real se proporciona al final de este estudio.

Términos del Índice—Blockchain, Sistema de votación, Voto electrónico

yo yoNTRODUCCIÓN

Un*elección*es un mecanismo central del sistema democrático. Un sistema de votación tradicional basado en papeletas es el enfoque adoptado con mayor frecuencia en una elección debido a la simplicidad de su implementación. Sin embargo, un sistema de votación basado en boletas de papel puede ser propenso a errores y también puede consumir mucho tiempo. Un sistema de votación ideal para una elección debe ser robusto, de modo que los errores humanos y los votos fraudulentos se minimicen, al mismo tiempo que garantiza que el proceso de recuento de resultados pueda completarse lo más rápido posible.

Para superar los desafíos mencionados, un*sistema de votacion electronica*se ha propuesto para resolver muchos de los problemas discutidos anteriormente. Hay varios sistemas de votación electrónica existentes que están listos para ser implementados [1]. Estos sistemas, sin embargo, aún no han sido ampliamente adoptados en una elección real, especialmente en una elección de alto perfil (por ejemplo, una elección política). La razón principal proviene del hecho de que los sistemas de votación electrónica existentes se basan principalmente en una arquitectura centralizada. Esto lleva a la posibilidad de que el resultado de la votación sea manipulado o manipulado mediante ciberataques o manipulación interna. Por lo tanto, un sistema de votación basado en boletas de papel sigue siendo preferible hoy en día porque no todos los votantes pueden confiar en un sistema de votación electrónico. Un sistema electrónico, en el que todos los votantes puedan confiar en que el resultado se mantiene seguro, confiable e inviolable, es necesario para ser propuesto

978-1-7281-2544-2/9/$31.00 ©2019 IEEE

En 2009, una tecnología financiera emergente llamada *Bitcoin*, ha sido propuesto por Satoshi Nakamoto [2]. Bitcoin permite que las transacciones financieras, que generalmente se procesan de manera centralizada, se procesen con un enfoque descentralizado a través del mecanismo subyacente denominado*cadena de bloques*. El mecanismo Blockchain permite a los usuarios realizar y registrar transacciones de manera distribuida, eficiente, permanente, no violable y verificable. Al adoptar Blockchain, los usuarios de Bitcoin pueden intercambiar efectivo directamente de igual a igual, mientras que se garantiza que cada transacción sea confiable sin depender de un control centralizado de un sistema bancario central. La capacidad de registrar transacciones globales de manera distribuida, pero segura y verificable, hace que los investigadores y los industriales hayan aplicado enfoques de Blockchain en muchos dominios de aplicaciones diferentes, sin limitarse solo a la aplicación financiera. Por ejemplo, un estudio en [3] aplica Blockchain en una aplicación para compartir datos médicos. Un estudio en [4] aplica Blockchain en una aplicación de monitoreo logístico. Sin embargo, hasta la fecha, el método adecuado, diseño,

En este estudio se identifican los requisitos para un adecuado sistema de votación electrónica de base descentralizada. La propuesta del diseño de la arquitectura y todos los modelos de datos relacionados de dicho sistema se describen a continuación. La arquitectura propuesta se denomina *BlockVOTE: Un Sistema de Votación Electrónica basado en Blockchain*. El detalle de la implementación de*BloqueVOTAR*también se da y el experimento de validación también se ha llevado a cabo.

La organización de este documento es la siguiente. La Sección II describe el escenario de motivación. El escenario seleccionado es un *elección política*. Los desafíos para proponer un sistema de votación de alta seguridad se presentan y analizan en relación con el escenario dado en esta sección. Esta sección sigue a la Sección III, que describe los estudios relacionados y el estado del arte de Blockchain. El BlockVOTE propuesto se describe en la Sección IV. Los experimentos de implementación y validación de BlockVOTE se describen en la Sección V. Los resultados y desafíos del experimento se analizan en la Sección VI. La sección VII concluye el estudio.

II. METROOTIVARSCENARIO: PAOLÍTICOmiLECCIÓN

Una elección es un proceso en el que la población elige de un grupo de personas para ocupar un cargo público, que a su vez representa las necesidades del público. Una autoridad electoral central es responsable de supervisar, organizar, validar y anunciar el resultado de la elección. La elección de cómo se organiza el sistema de votación puede variar dependiendo de la elección de la autoridad electoral de cada organización o país. En general, hay dos tipos principales de sistemas de votación, que son (i) un sistema de votación basado en boletas y (ii) un sistema de votación electrónico. Se describen a continuación.

A. Sistemas de votación basados en boletas

El sistema de votación centralizado más común es el sistema de boletas. La autoridad electoral determinaría la lista de votantes potenciales en base a criterios predeterminados. Cada uno de los votantes recibe una boleta de papel, y luego el votante selecciona una de las opciones posibles antes de depositar su boleta en una urna en un colegio electoral.

Los sistemas de votación tradicionales basados en boletas tienen muchas desventajas. La preparación de papeletas de votación en papel es una tarea costosa y que consume mucho tiempo, en la que las papeletas de papel deben prepararse de antemano para dar cuenta de todos los votantes potenciales de cada centro de votación y no son reutilizables. Otro tema importante es el largo proceso de conteo de votos. El proceso de conteo de votos requiere mucho tiempo ya que todas las boletas deben ser examinadas, clasificadas y contadas. Además, el proceso de conteo puede alterarse si las autoridades electorales son parciales y/o ninguna parte interesada estuvo monitoreando el proceso de conteo para verificarlo dos veces.

B. Sistemas de votación electrónicos

Para mejorar el sistema de votación con boletas en papel, los sistemas de votación electrónicos que utilizan *Máquina de votación electrónica (EVM)* [5], que pueden diferir en los detalles de implementación. Los ejemplos incluyen los sistemas EVM que proporcionan impresión y marcado de boletas a pedido, sistemas que pueden contar la votación, sistemas que permiten la transmisión de resultados tabulados y sistemas de votación en línea.

Los sistemas de votación en línea ofrecen la mejor solución cuando se considera la eficiencia en los sistemas de votación, pero existen numerosos problemas debido a su diseño centralizado que hace que el sistema sea vulnerable a ciberataques externos y manipulación por parte de las autoridades electorales. La infraestructura de la votación debe diseñarse de tal manera que todos los resultados de la votación deben mantenerse protegidos contra amenazas externas en línea y manipulación interna.

C. Desafíos: un sistema de votación de alta seguridad

En un caso ideal, todas las elecciones prefieren un sistema de votación que garantice un alto nivel de seguridad ya prueba de manipulaciones, mientras que el proceso de conteo y el anuncio de los resultados se mantienen para que sean lo más eficientes y lo antes posible. Para hacerlo, resumimos los desafíos que deben abordarse de la siguiente manera.

- *Protección de la infraestructura de red:* Un sistema de votación electrónica altamente seguro debe garantizar que todos los dispositivos relacionados con la infraestructura de la red puedan resistir los ataques cibernéticos;
- *Protección de los datos de votación:* Un sistema de votación electrónica de alta seguridad debe mantener todos los datos de la votación de tal manera que el resultado siempre pueda estar disponible para el conteo y garantizar que todos los datos no puedan ser manipulados mediante ciberataques ni manipulaciones internas;
- *Gestión de confianza:* Un sistema de votación electrónica altamente seguro donde los resultados son 100% confiables y confiables para todas las partes relacionadas.

tercero REXALTADOSESTUDIOS: BLOCKCHAIN

cadena de bloques es un algoritmo que maneja transacciones entre libros de igual a igual. Inicialmente, está diseñado para admitir el procesamiento de transacciones en Bitcoin [2]. Bitcoin es un sistema bancario descentralizado en el que cada transacción entre libros de contabilidad se puede realizar directamente sin depender de un servidor centralizado o una autoridad bancaria central para validar las transacciones. Los libros de contabilidad en el ecosistema de Bitcoin intentan validar la validez de la transacción por sí mismos, en los que cada transacción dentro del ecosistema se modela como un *Bloquear*. Un bloque recién creado solo se puede conectar a la cadena global compartida de bloques solo si todos los registros dentro del ecosistema de Bitcoin validan el bloque para que sea válido. Este proceso ayuda a Bitcoin a mantener todas las transacciones altamente seguras, inviolables y a prueba de manipulaciones sin depender de un banco central para validar una transacción [6].

El avance en Blockchain se produjo en la forma en que el bloque se puede incrustar con una función programable incorporada para admitir una lógica comercial personalizada. Tal extensión se considera como un *Contrato inteligente* característica de *cadena de bloques*. A *contrato inteligente* es un proceso autónomo que es capaz de regular el flujo de transacciones dentro de una red Blockchain, de modo que se ejecutaría un conjunto específico de instrucciones programables que se acuerdan en cada libro mayor cada vez que se realizaran una acción o eventos predefinidos dentro de la red Blockchain [7].

Blockchain y Smart Contract se consideran adecuados para almacenar conocimiento global y lógica comercial para cualquier dominio de aplicación de manera distribuida, manteniendo la seguridad y la privacidad del conocimiento almacenado. Esto permite una gama más amplia de dominios de aplicación. Los ejemplos de dominios de aplicación que utilizan Blockchain incluyen el almacenamiento de registros médicos [8], la gestión de la cadena de suministro [9] y los sistemas de autenticación [10].

IV. BCERRARVOTO: UNnorteAARQUITECTURA DE UN BLOCKCHAIN-ESTABLECIDOmiLECTRONICAVOTARSYSTEM

En este estudio, proponemos una arquitectura de un sistema de votación electrónica basado en blockchain. Nuestra arquitectura se llama *BloqueVOTAR*. Esta sección describe la descripción general de la arquitectura, los modelos de datos, los algoritmos y todas las definiciones formales relacionadas. Los detalles sobre la implementación de nuestra propuesta se describen más adelante en la siguiente sección. la arquitectura de *BloqueVOTAR* se representa en la Figura 1. Nuestra arquitectura está diseñada en base a la *contrato inteligente* capacidad de Blockchain. Para mayor claridad de la explicación, explicamos nuestra arquitectura paso a paso de acuerdo con los tres pasos principales de cualquier votación.

proceso, que son (i) creación de encuestas, (ii) votación y (iii) conteo de resultados. Los detalles son los siguientes.

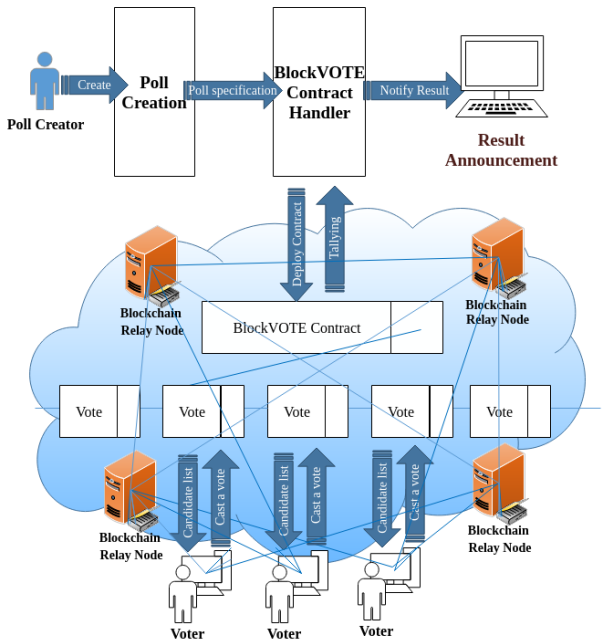

Figura 1. BlockVOTE: descripción general de la arquitectura

A. Creación de encuestas

Antes de crear una encuesta, primero se necesitaría preparar una lista de candidatos. El proceso interno de elección de candidatos (por ejemplo, verificar el perfil de cada candidato en una elección política) puede variar entre cada organización. Este proceso no está cubierto por nuestra arquitectura, ya que se considera fuera del alcance del sistema. Cuando la lista de candidatos esté lista, el creador de la encuesta debe enviar la lista de candidatos al *Creación de encuestas* (ver Fig. 1) módulo de la arquitectura. Este módulo se utiliza para modelar la lista de candidatos en un formato legible por máquina. El modelo de datos utilizado para modelar un candidato se describe en Def 1.

Definición 1: Candidato: A *Candidato*, C, es una opción dentro de una encuesta a la que un votante puede votar. Se modela como una tupla de 3 $C = < id, descripción, puntuación >$, dónde:

- *identificación*: es un número único para identificar C;
- *descripción*: es una breve descripción textual fácil de usar para describir C;
- *puntaje*: es un número que representa el número de votos que C obtuvo de los votantes.

Después de modelar todos los candidatos, el *Creación de encuestas* pasaría automáticamente la lista de candidatos modelados al *Manejador de contratos BlockVOTE* módulo. *Manejador de contratos BlockVOTE* es responsable de crear un *Contrato BlockVOTE*, que es un contrato inteligente diseñado específicamente para la arquitectura propuesta. La definición formal de la *Contrato BlockVOTE* se da en Def. 2.

Algoritmo 1 Votar: una función incorporada de BlockVOTE

Requerir:

BV: un contrato BlockVOTE que un votante invoca

identificación: un número de identificación de un candidato que un votante que invoca el contrato quiere votar

v: un hash criptográfico basado en blockchain de un votante que invoca *BV*

si v es en $BV.V$ después
    devolver *Falso* . v ya ha emitido un voto
más sininguno de $c.id$ en $BV.C$ es igual a *identificación* después
    devolver *Falso* . ID de candidato no válido
más
    $c.puntuación = c.puntuación + 1$ dónde $C \in BV.C \wedge c.id = identificación$
    agregar v a $BV.V$
    devolver *Verdadero* . Emitir con éxito un voto
terminara si

Definición 2: Contrato BlockVOTE: A *Contrato BlockVOTE*, *BV*, es un contrato inteligente que es responsable de administrar una encuesta, manejar las actividades de votación y contar el resultado automáticamente. Se define como una tupla de 3, $BV = < C, V, Exp >$ mientras:

- C: es un conjunto de candidatos definido como $C = \{C_1, C_2, C_3, ..., C_{norte}\} \wedge \forall C_i \in do, do = < i, desc, partitura >$ (ver Definición 1);
- V: es un conjunto de votantes que han emitido un voto a uno de los candidatos en C, $V = \{v_1, v_2, v_3, ..., v_{norte}\} \wedge \forall v_i \in v_i$ es un hash criptográfico basado en blockchain que representa la identidad de un votante determinado;
- *Exp*: es un valor de fecha y hora que indica que cuando los votantes aún pueden emitir un voto para C;

Cuando *Manejador de contratos BlockVOTE* recibe una solicitud de creación de encuesta, crea un nuevo contrato BlockVOTE de acuerdo con Def 2, manteniendo V vacío, y ajuste *Exp* de acuerdo con la solicitud de un creador de encuestas. Cuando se crea con éxito un nuevo contrato, se implementa automáticamente en una infraestructura basada en Blockchain implementada previamente en la nube de Internet de tal manera que los usuarios pueden votar por el contrato. El detalle de la *votar* funcion de *BV* se describe en la siguiente subsección.

B. Votación

Después de que el creador de la encuesta use *Creación de encuestas* módulo y *Manejador de contrato Block-VOTE* módulo para crear un nuevo contrato BlockVOTE, los votantes pueden invocar el *votar* funcionalidad del contrato para emitir un voto. La funcionalidad se define como una función integrada que utiliza *BV* como entrada. el algoritmo de la *votar* La función se describe en Alg. 1. En resumen, un votante emite un voto al pasar (i) un contrato para votar, (ii) una identificación de un candidato que un votante quiere, y (iii) su propia identidad hash criptográfica, al *votar* función. La votación es un éxito solo si ese votante aún no ha emitido un voto, y siempre que la identificación del candidato sea válida. La función de voto rechaza un voto cuando un votante ya ha emitido un voto, o cuando la identificación del candidato proporcionada no es válida.

Cada vez que se ha emitido un voto usando una función como se describe en Alg. 1, un nuevo *Bloquear* es creado. Un bloque dentro de la arquitectura BlockVOTE almacena una transacción de voto de un votante y el estado más reciente del contrato (es decir, una lista de todos los candidatos y su puntaje correspondiente recibido).

C. Conteo de resultados

El cómputo de resultados es un proceso que se realiza cuando se ha llegado a la fecha límite de la encuesta. El proceso de conteo se modela como una función basada en un contrato inteligente que ejecuta automáticamente el *Manejador de contratos BlockVOTE* módulo cuando se alcanza la fecha límite de la encuesta. La definición formal de la función de conteo de resultados se da en Def. 3.

Definición 3: Función de conteo de resultados: Una función de conteo de resultados, *t.f.*, es una función para contar el resultado de una encuesta dada. se define como

$$t.f.(BV, cfecha) = \begin{cases} falso & \text{si } cfecha < BV.exp \\ R & \text{de lo contrario} \end{cases}$$

mientras:

- *BV*: es un contrato BlockVOTE para estar contando *t.f.*, $BV = < C, V, Exp >$ (ver Def. 2);
- *cfecha*: es una fecha-hora cuando se solicita un proceso de conteo.
- *R*: es un conjunto ordenado, $R = \{C_1, C_2, C_3, ..., C_{norte}\} \wedge \forall C_i \in R, c_i \in BV.C \wedge C_i.puntuación < c_{i+1}.puntaje$

En resumen, la función de conteo de resultados devuelve una lista de candidatos dentro de un contrato ordenados por el puntaje que recibe cada candidato en orden descendente. La función devuelve *falso* to indica que no se puede hacer el conteo si la encuesta aún no ha terminado.

Hasta ahora, la arquitectura se ha descrito utilizando todas las definiciones y algoritmos formales relacionados. En la siguiente sección se describe la implementación de la arquitectura propuesta.

VBCERRARVOTO: YOIMPLEMENTACIÓN& VALIDACIÓN

Para garantizar la viabilidad del diseño de la arquitectura propuesta, la arquitectura propuesta debe ser lo suficientemente genérica para desarrollarse con diferentes marcos de aplicación de Blockchain. Se desarrollaron dos prototipos separados para el sistema propuesto. El primer prototipo fue desarrollado utilizando *Etéreo* estructura[1]. El otro prototipo fue desarrollado usando *Hiperlibro mayor* estructura[2]. Esta sección describe el detalle de implementación y el experimento de validación de los prototipos.

A. BlockVOTE: Implementación basada en Ethereum

Ethereum es uno de los ecosistemas de criptomonedas más famosos que admite una capacidad de contrato inteligente, de manera que cualquier propietario de billetera Ethereum puede implementar su propio contrato inteligente en la red Ethereum a nivel mundial para respaldar su propia lógica comercial. Implementar BlockVOTE en Ethereum, cumpliendo con el marco de desarrollo de contratos inteligentes de la fundación Ethereum llamó a la *Marco de trufa*[3] fue requerido. La arquitectura de la implementación BlockVOTE basada en Etherum se muestra en la Figura 2.

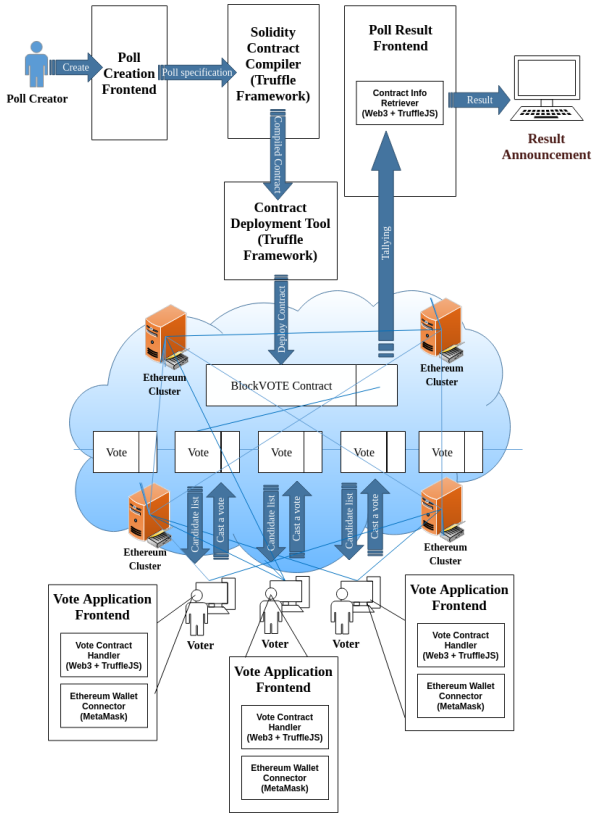

Fig. 2. BlockVOTE: Implementación basada en Ethereum

Para el prototipo propuesto, como se muestra en la Figura 2, el creador de la encuesta creará una encuesta utilizando una interfaz de aplicación basada en web. Esta aplicación frontend toma la lista de candidatos proporcionada por los usuarios para generar un código fuente de contrato inteligente que luego sería utilizado por Ethereum. Ethereum especifica que un contrato inteligente debe estar escrito en *Solidez* idioma. la plantilla de la *BloqueVOTAR* El contrato en el lenguaje de Solidity se define de la siguiente manera:

```
solidez de pragma ^0.5.0;
contrato Bloquear voto {
    Candidato de estructura {
        ID de unidad;
        descripción de cadena;
        puntuación unitaria;
    }
    mapeo(dirección => bool) votantes públicos; mapeo(uint
    => Candidato) candidatos públicos; uint candidaturas
    públicasCount;
    uint256 public exp_date;

    función addCandidate (cadena de memoria _name) privado {
        candidatosCount ++;
        candidatos[candidatesCount] = Candidate(candidatesCount, _name, 0);
    }
    función votar (uint _candidateId) público {
        require(!votantes[mensaje.remitente]);
        require(_candidateId > 0 && _candidateId <= candidatasCount);
        votantes[msg.sender] = verdadero;
        candidatos[_candidateId].puntuación++;
    }
}
```

Después de generar el código del contrato, se compilaría utilizando la herramienta de compilación Solidity proporcionada por Truffle.

[1] https://www.ethereum.org/
[2] https://www.hyperledger.org/
[3] https://truffleframework.com

Estructura. El contrato compilado se implementaría más tarde en la red Ethereum mediante el uso de la herramienta de migración de contratos.

Cada votante necesitaría tener acceso a una máquina de votación que podría ser una PC, un teléfono inteligente o una tableta que esté instalada con la aplicación y tenga capacidad para Internet. La aplicación utiliza Web3[4]y la biblioteca TruffleJS para interactuar con el contrato BlockVOTE implementado.

Aunque la votación debería ser gratuita, la especificación de la plataforma Ethereum requiere que todos los usuarios gasten una cierta cantidad de*Éter*(el nombre de la moneda en el ecosistema Ethereum) antes de que se les permita crear un nuevo bloque. La cantidad de Ether que un usuario necesita gastar depende del tamaño de los datos dentro del bloque creado. Por lo tanto, en la aplicación de votación propuesta basada en Ethereum, cada usuario debe conectar su propia billetera Ethereum a la aplicación de votación antes de poder emitir un voto. El costo por voto en la aplicación BlockVOTE, según el cálculo automático de costos en la red Ethereum, es de 0.000652 ETH por voto.

Para el proceso de conteo de resultados, se creó una interfaz separada utilizando las bibliotecas Web3 y TruffleJS. El proceso de conteo se realizó enumerando todos los candidatos y sus puntajes como se indica en el contrato en la interfaz. La interfaz rechazará la solicitud de conteo si la encuesta aún no ha finalizado de acuerdo con la fecha de vencimiento de la encuesta indicada en el contrato.

B. BlockVOTE: Implementación basada en HyperLedger

HyperLedger es una plataforma de cadena de bloques de código abierto alojada por Linux Foundation. A diferencia de Ethereum, el diseño de HyperLedger no está diseñado para crear un ecosistema público basado en criptomonedas. En cambio, fue diseñado para desarrollar un ecosistema de cadena de bloques privado donde las organizaciones privadas pueden crear su propia red de cadena de bloques internamente e implementar su lógica comercial en su red utilizando un mecanismo inteligente basado en contratos. La descripción general de la arquitectura de la implementación BlockVOTE basada en HyperLedger del sistema propuesto se muestra en la Figura 3.

La arquitectura de las implementaciones de BlockVOTE Hyperledger es similar a la implementación de Ethereum. Las diferencias entre ambas implementaciones son los lenguajes para modelar un contrato y el conjunto de herramientas requeridas. En Hyper-Ledger, los modelos de datos y las funciones integradas se programan por separado. Los modelos de datos se escriben utilizando un lenguaje específico de HyperLedger llamado*CTO*lenguaje, mientras que las funciones integradas se escriben usando Javascript. La plantilla que utiliza la arquitectura propuesta para generar un modelo de datos y la función integrada para crear un nuevo contrato BlockVOTE son las siguientes:

```
//Plantilla de modelo de datos// espacio de
nombres org.acme.blockvote

votante participante identificado por voterID {
   o Cuerda   ID de votante
   o Cuerda   nombre completo
}

activo ifVoted identificado por voterID {
   o ID de votante de cadena
   o Booleano esvoto
}

activo candidatoVoto identificado por candidatoID {
   o Cuerda   Identificación del candidato
   o Cadena short_desc
   o Voto total entero
}

voto de transacción {
   - - >  candidatovotar    candidatoVoteActivo
   - - >  si votó si votó activo
}

//Guion      Modelo//
'uso estricto';

función votar(tx) {
      if (!tx.ifVotedAsset.isvote)                {
            tx.candidateVoteAsset.totalVote = tx.candidateVoteAsset.totalVote + 1; devuelve
            getAssetRegistry('org.acme.blockvote.candidateVote')
                  . entonces (función (registro de activos) {
                        volver assetRegistry.update(tx.candidateVoteAsset)
                           . entonces(función(){
                                 devuelve getAssetRegistry('org.acme.blockvote.candidateVote')
                              })
                  })
                  . entonces (función () {
                        devuelve getAssetRegistry('org.acme.blockvote.ifVoted')
                              . luego (función (registro de activos)
                                    { tx.ifVotedAsset.isvote = true;
                                    return assetRegistry.update(tx.ifVotedAsset);
                                 })
                  });
      } más {
            throw new Error('¡Voto ya enviado!');
      }
}
```

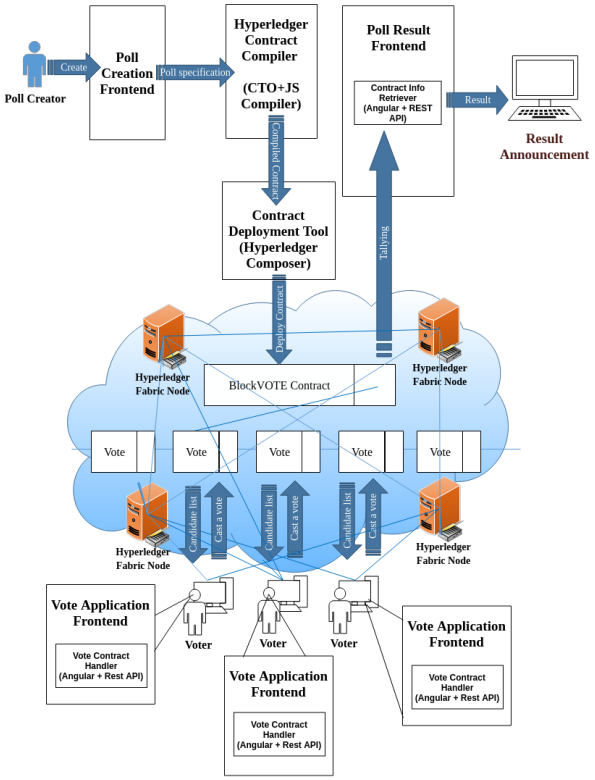

Fig. 3. BlockVOTE: Implementación basada en HyperLedger

El contrato que ha sido generado sería compilado y desplegado usando*Compositor de HyperLedger*conjuntos de herramientas El contrato BlockVOTE compilado luego se implementaría en una red de un*Tejido HyperLedger*[5]nodos. Este paso es obligatorio cuando se usa HyperLedger, ya que HyperLedger es una plataforma privada basada en blockchain. Por lo tanto, un conjunto de nodos

[4]https://web3js.readthedocs.io/en/1.0/

[5]HyperLedger Fabric es un software que se utiliza para crear un nodo de servidor blockchain para HyperLedger

son necesarios para crear la infraestructura para HyperLedger. Este paso no era necesario en el prototipo basado en Ethereum, ya que la red Ethereum ya tiene disponible una extensa red de clústeres de nodos Ethereum.

Para emitir un voto, los votantes deben tener una aplicación de interfaz para emitir un voto. La aplicación frontend fue desarrollada usando Angular[6]y un conjunto de API REST basada en la web proporcionada por HyperLedger Fabric para conectarse al nodo de HyperLedger Fabric más cercano. El prototipo de HyperLedger no requiere una billetera de criptomonedas porque HyperLedger asume que el propietario de la aplicación posee toda la infraestructura relacionada. Por lo tanto, los votantes no necesitan pagar ninguna tarifa para emitir un voto en la implementación de la arquitectura basada en HyperLedger, lo cual es una ventaja considerable sobre la implementación basada en Ethereum.

C. Experimento de Validación

Para validar el sistema propuesto, se creó e implementó una encuesta simulada con 10 opciones de candidatos tanto en los prototipos basados en Ethereum como en HyperLedger.

El desempeño del proceso de conteo sería crucial para una aplicación de votación. Se realizaron tres conjuntos de experimentos con 10, 50 y 100 votos. Para cada conjunto, se llevan a cabo cinco rondas de experimentos y el retraso de tiempo promedio entre el momento en que se solicitó la función de conteo de resultados y cuando el resultado está disponible en la notificación de resultados de la interfaz. El resultado se muestra en la Figura 4.

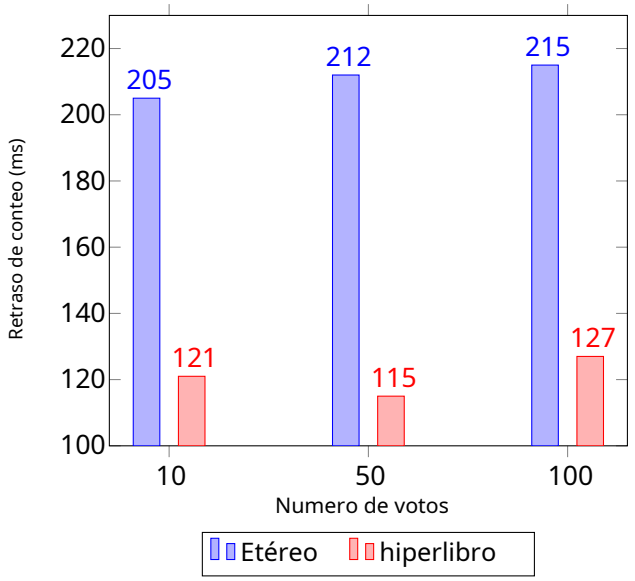

Fig. 4. Retraso en el recuento de la implementación de Block-VOTE basada en Ethereum e HyperLedger

La Figura 4 demuestra que el retraso de conteo para todos los casos es casi constante para cada implementación respectiva. La razón proviene del hecho de que cada bloque en la red siempre contiene el último estado del contrato. Por lo tanto, el proceso de conteo no requiere acceder a todos los datos. desde el principio hasta el final de la cadena. Solo se requiere el último bloque para el conteo de resultados.

También se puede observar a partir del resultado que el proceso de conteo de HyperLedger es ligeramente más rápido que Ethereum. Esto proviene de la diferencia en el algoritmo de manejo de consenso interno entre Ethereum e HyperLedger. Ethereum adopta un consenso basado en Prueba de participación que requiere que los usuarios paguen a Ether para garantizar una transacción [11]. Hyper-Ledger utiliza un algoritmo de manejo de consenso basado en certificados a través del consenso de prueba de autoridad [12] usando la biblioteca Apache Kafka[7]. El consenso de prueba de autoridad funciona más rápido que el consenso de prueba de participación, lo que conduce a un retraso más rápido en el proceso de conteo cuando se implementa BlockVOTE en HyperLedger. Sin embargo, la Prueba de participación tiene un mecanismo de manejo de consenso más seguro [13] que la Prueba de autoridad según muchos investigadores y desarrolladores de Blockchain.

VI. Discusión

Desafíos técnicos significativos, como se menciona en la Sección II, están relacionados con la seguridad de la infraestructura de red y los datos de votación. Para superar estos desafíos, este estudio propone un sistema de votación electrónica basado en descentralización utilizando Blockchain. El sistema propuesto almacena datos de votación entre los nodos de blockchain descentralizados. La manipulación de los resultados de la votación en uno de los nodos ya no es posible porque otros nodos no aceptarían el resultado manipulado a través del mecanismo de manejo de consenso de Blockchain. Por lo tanto, los desafíos para asegurar tanto la infraestructura como los datos de votación podrían superarse.

Se crearon prototipos utilizando la plataforma Ethereum y HyperLedger para realizar una encuesta, y los resultados se contaron en el experimento de validación. Según el experimento, uno de los principales problemas a considerar es que el prototipo basado en Ethereum requiere que los votantes posean una billetera Ethereum antes de poder emitir un voto, mientras que no existe tal requisito para el prototipo basado en HyperLedger. Esto se debe al hecho de que implementar BlockVOTE en Ethereum significa que todos los datos de votación deben almacenarse en la red pública de Ethereum y todos los votantes deben seguir el*Prueba de participación*protocolo de consenso de Ethereum pagando una cierta cantidad de Ether antes de que se pueda emitir un voto. Esto se considera una desventaja considerable al adoptar Ethereum sobre HyperLedger porque no es viable obligar a todos los votantes a poseer una billetera Ethereum y poseer créditos Ethereum.

Por otro lado, implementar BlockVOTE usando Hyper-Ledger no requiere que los votantes posean una billetera de criptomonedas. Sin embargo, la principal desventaja de adoptar HyperLedger es que requiere la implementación de varios nodos de HyperLedger Fabric antes de que sea posible desarrollar e implementar cualquier aplicación. La implementación es necesaria ya que no existe una extensa red pública gratuita de HyperLedger.

Los desarrolladores que busquen implementar una aplicación de votación electrónica en Blockchain deberán sopesar estas limitaciones mencionadas de Ethereum e HyperLedger antes de comenzar cualquier implementación.

[6]https://angular.io/

[7]https://kafka.apache.org/

VIII. Conclusión y Futura Work

Aunque los sistemas de votación centralizados tradicionales se han utilizado desde la invención de la democracia, existen muchas posibilidades para mejorar los sistemas de votación. En el área de los sistemas de votación centralizados, se ha explorado el uso de sistemas de votación en línea que ofrecen una mejora convincente de los sistemas de votación y los sistemas de votación electrónica. Sin embargo, los sistemas de votación en línea son complejos y el sistema tiene problemas de seguridad de los que se podría abusar y tendría graves ramificaciones en los resultados de las elecciones.

Un enfoque alternativo en el sistema de votación es explorar otros enfoques. La tecnología Blockchain, que es un libro mayor abierto, distribuido y autoauditado que puede registrar transacciones entre partidos de manera eficiente, permanente y verificable, es una tecnología que podría adaptarse para su uso en sistemas de votación. Por lo tanto, en este estudio, el *BlockVOTE: una arquitectura de un sistema de votación electrónica basado en blockchain* fue propuesto. En este estudio se propusieron la arquitectura, los modelos de datos y todas las definiciones formales relacionadas. La arquitectura se validó empleando prototipos y experimentando con dos marcos de implementación diferentes, que son Ethereum e HyperLedger. El resultado muestra que el sistema propuesto, en ambas implementaciones prototipo, podría usarse para realizar una encuesta, manteniendo el resultado seguro y minimizando el tiempo de conteo de resultados.

Aunque nuestro sistema de votación propuesto puede proporcionar muchas ventajas, es necesario abordar numerosos problemas antes de la adopción más amplia de la tecnología. Por lo tanto, para el trabajo futuro, hay varios temas que el equipo está abordando. La primera área es finalizar el desarrollo y desplegar el prototipo para una votación a gran escala en lugar de los experimentos de prueba limitados. Un experimento más masivo con votantes en vivo podría propagar la utilidad de la tecnología Blockchain en las elecciones a una audiencia más amplia para que las autoridades electorales y las organizaciones gubernamentales examinen la tecnología y puedan adoptarla para votaciones futuras. Otra dirección es explorar la propuesta de un marco de aplicación basado en Blockchain que sea más adecuado para ser utilizado en un sistema de votación electrónica que Ethereum o HyperLedger.

Rferencias

[1] MK Alomari, “Adopción del voto electrónico en un país en desarrollo”, *Transformación del gobierno: personas, procesos y políticas*, vol. 10, núm. 4, págs. 526–547, 2016. [En línea]. Disponible: https://doi.org/10.1108/TG-11-2015-0046

[2] S. Nakamoto, "Bitcoin: un sistema de efectivo electrónico entre pares, http://bitcoin.org/bitcoin.pdf", 2009.

[3] Q. Xia, EB Sifah, KO Asamoah, J. Gao, X. Du y M. Guizani, "Medshare: intercambio de datos médicos sin confianza entre proveedores de servicios en la nube a través de blockchain".*Acceso IEEE*, vol. 5, págs. 14 757–14 767, 2017.

[4] R. Casado-Vara, A. González-Briones, J. Prieto y JM Corchado, “Contrato inteligente para el seguimiento y control de actividades logísticas: estudio de caso de empresas farmacéuticas”, en *Conferencia Internacional Conjunta SOCO'18-CISIS'18-ICEUTE'18*, M. Graña, JM López-Guede, O. Etxaniz, Á. Herrero, JA Sáez, H. Quintián y E. Corchado, eds. Cham: Springer International Publishing, 2019, págs. 509–517.

[5] T. Kohno, A. Stubblefield, AD Rubin y DS Wallach, “Análisis de un sistema de votación electrónica”, en *Simposio IEEE sobre seguridad y privacidad, 2004. Actas. 2004*, mayo de 2004, págs. 27–40.

[6] A. Gervais, GO Karame, K. Wüst, V. Glykantzis, H. Ritzdorf y S. Capkun, "Sobre la seguridad y el rendimiento de las cadenas de bloques de prueba de trabajo", en *Actas de la Conferencia ACM SIGSAC 2016 sobre seguridad informática y de las comunicaciones*, ser. CCS'16. Nueva York, NY, EE. UU.: ACM, 2016, págs. 3–16. [En línea]. Disponible: http://doi.acm.org/10.1145/2976749.2978341

[7] H. Watanabe, S. Fujimura, A. Nakadaira, Y. Miyazaki, A. Akutsu y J. Kishigami, "Contrato de cadena de bloques: asegurar una cadena de bloques aplicada a contratos inteligentes", en *2016 Conferencia internacional IEEE sobre electrónica de consumo (ICCE)*, enero de 2016, págs. 467–468.

[8] A. Roehrs, CA da Costa y R. da Rosa Righi, "Omniphr: un modelo de arquitectura distribuida para integrar registros de salud personales" *Revista de informática biomédica*, vol. 71, págs. 70 – 81, 2017. [En línea]. Disponible: http://www.sciencedirect.com/science/article/pii/S1532046417301089

[9] D. Tse, B. Zhang, Y. Yang, C. Cheng y H. Mu, "Blockchain application in food supply information security", en *Conferencia internacional IEEE 2017 sobre ingeniería industrial y gestión de ingeniería (IEEM)*, diciembre de 2017, págs. 1357–1361.

[10] J.-H. Huh y K. Seo, "Plataforma de inicio de sesión automático y verificación de huellas dactilares móviles basada en blockchain para la informática del futuro" *El diario de la supercomputación*, vol. 75, núm. 6, págs. 3123–3139, junio de 2019. [En línea]. Disponible: https://doi.org/10.1007/s11227-018-2496-1

[11] W. Li, S. Andreina, J.-M. Bohli y G. Karame, "Asegurar los protocolos de cadena de bloques de prueba de participación", en *Gestión de privacidad de datos, criptomonedas y tecnología Blockchain*, J. Garcia-Alfaro, G. Navarro-Arribas, H. Hartenstein, and J. Herrera-Joancomartí, Eds. Cham: Springer International Publishing, 2017, págs. 297–315.

[12] E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. De Caro, D. Enyeart, C. Ferris, G. Laventman, Y. Manevich, S. Muralidharan, C. Murthy, B. Nguyen, M. Sethi, G. Singh, K. Smith, A. Sorniotti, C. Stathakopoulou, M. Vukolić, SW Cocco y J. Yellick, "Hyperledger fabric: A distribution operating system for allowed blockchains", en *Actas de la Decimotercera Conferencia EuroSys*, ser. EuroSys'18. Nueva York, NY, EE. UU.: ACM, 2018, págs. 30:1–30:15. [En línea]. Disponible: http://doi.acm.org/10.1145/3190508.3190538

[13] L. Ismail, H. Hameed, M. AlShamsi, M. AlHammadi y N. AlDhanhani, "Hacia un despliegue de blockchain en la universidad de los EAU: evaluación del rendimiento y taxonomía de blockchain", en *Actas de la Conferencia Internacional sobre Tecnología Blockchain de 2019*, ser. ICBCT 2019. Nueva York, NY, EE. UU.: ACM, 2019, págs. 30–38. [En línea]. Disponible: http://doi.acm.org/10.1145/3320154.3320156