Lamden: un entorno para el desarrollo y despliegue rápido de aplicaciones para *Blockchain*.

Stuart Farmer[[1]](#footnote-1)

Mario Hernandez[[2]](#footnote-2)

James Munsch[[3]](#footnote-3)

El desarrollo de aplicaciones para Blockchain es innecesariamente difícil, en parte debido a lo novedoso de la tecnología, pero también debido a ciertas limitaciones inherentes al status quo actual de la utilización de las blockchains. En este artículo, discutimos los problemas de las actuales implementaciones blockchain y una solución a estos problemas. Presentamos Lamden: un entorno de desarrollo que incluye un generador de blockchains, herramientas de colaboración, y un protocolo de comunicación entre blockchains para crear extensas redes que puedan transferir cualquier tipo de información a través de ellas.

**Palabras clave:** descentralización, criptomoneda, sistemas distribuidos

# Motivación

En los últimos años, hemos presenciado cómo la tecnología basada en la blockchain golpeaba el mercado. Estas tecnologías se consideran como un medio para el fin de los sistemas bancarios y financieros centralizados. Sin embargo, estas cadenas dependen de sistemas defectuosos que no tienen en cuenta la aplicación de dichas tecnologías en el marco de la infraestructura actual. Como resultado, el avance en la tecnología de las blockchains está segregada de los sistemas financieros arquetípicos, y por tanto obstaculizada a la hora de lograr su objetivo.

El Proyecto Lamden es un intento de crear herramientas de desarrollo de software que promuevan la adopción a gran escala de la tecnología blockchain para que pueda revolucionar los sistemas de mercado y causar innovación real dentro de estos.

En este documento se exponen nuestras críticas al estado actual del mercado, y nuestros planes para mejorar este espacio.

# Problemas de las implementaciones actuales de la Blockchain

Hay muchos defectos de base en la arquitectura blockchain que le impiden convertirse en una tecnología convencional. Estos defectos existen tanto de cara al usuario como en el lado del desarrollador / proveedor de servicios. Estas son las áreas en las que esperamos mejorar las implementaciones actuales de la Blockchain con este proyecto.

**Tasas sobre las transacciones**

En primer lugar, vamos a acercarnos al problema con las tasas de transacción desde el punto de vista del usuario final. La clave del éxito en el espacio de la criptodivisa es la adopción masiva de Bitcoin en el mercado internacional. En el momento de la redacción de este documento, la tasa de transacción promedio para una transacción de Bitcoin ronda el 6%. Sin embargo, comparado con sistemas centralizados como PayPal, Mastercard o Visa, una transacción realizada con Bitcoin acumula el doble de tasas.

Cabe señalar que, para el usuario medio, los fundamentos de la criptomoneda son el valor añadido que compensa estas discrepancias. La naturaleza descentralizada de la divisa supera los costos adicionales. Pese a todo, a escala macroeconómica la rentabilidad superará el atractivo fundamental.

*“Y Ethereum? Sus tasas de transacción son sólo del 0.3%!”*

Si bien esto es cierto, Ethereum no fue diseñado como una red de transferencia de valor como lo fue Bitcoin, sino más bien una red de contratos inteligentes (smart contracts) para permitir sistemas de software descentralizados y de confianza. Tenemos que recordar que con Ethereum, los usuarios pagan por la cantidad de operaciones que se realizan en la cadena, y que ésta sigue aumentando a medida que se despliegan contratos inteligentes más complejos. Una de las ventajas más importantes de Ethereum es la capacidad de agregar contratos inteligentes como "bibliotecas" que funcionan de la misma manera que en otros entornos de desarrollo de software, automatizando operaciones y procesos para que los usuarios no tengan que volver a escribirlos una y otra vez. Como tal, podemos asumir que, con el tiempo, a medida que la complejidad de estos contratos crece y se integran más bibliotecas, el costo de interactuar con la cadena principal de Ethereum también se incrementará. Este mecanismo basado en cobrar a los desarrolladores sobre las instrucciones desincentiva la innovación y los complejos sistemas de contratos inteligentes necesarios para el éxito a largo plazo de Ethereum.

**Cadenas sobrecargadas**

Continuando con Ethereum, dado que el libro de contabilidad (*ledger*) es público, cualquiera puede implementar el contrato inteligente que desee. Como resultado, a pesar de que la cadena tiene sólo dos años, el tamaño de un nodo completo es de 242 GB, que es incluso mayor que el tamaño actual de un nodo completo Bitcoin, de alrededor de 152 GB. Tener que instalar un nodo completo, o al menos uno ligero (que todavía supone en torno a 20 GB de datos) representa un obstáculo mayor para alguien que sólo quiere probar el desarrollo de contratos inteligentes.

Compare el tiempo que se tarda en sincronizar completamente un nodo de Ethereum con el tiempo que se tarda en instalar Python, Node.js, o cualquier otro entorno de programación para comenzar a explorar. La inconveniencia y el tamaño de la descarga por sí solos son inviables para la mayoría de los desarrolladores y empresas, y mucho menos para el usuario promedio que tiene curiosidad por la tecnología Blockchain.

**Escalabilidad y volumen de transacciones**

Las cadenas actuales tienen un rendimiento notoriamente bajo, lo que hace que los tiempos de transacción sean lentos, lo que se traduce en frecuentes acumulaciones de transacciones pendientes. Esto es algo que los sistemas financieros no están dispuestos a aceptar, con millones de transacciones procesadas cada día.

Por ejemplo, Bitcoin tiene un ancho de banda de 3 transacciones por segundo. Ethereum llega hasta las 15 transacciones por segundo. En comparación, PayPal lanzó un estudio en mayo de 2015, demostrando la capacidad de su sistema para gestionar más de mil millones de transacciones al día, lo que se traduce en unas 11.600 transacciones por segundo.

Todo esto disuade a las empresas que gestionan sus datos con bases de datos tradicionales de alto rendimiento, con las que consiguen altas disponibilidades del sistema.

**Minado y sostenibilidad**

Otra barrera para la adopción masiva son las funciones de hashing arbitrarias que los sistemas de verificación utilizan para asegurar la integridad de la cadena. Estas máquinas utilizan cantidades masivas de recursos en términos de energía que no son razonables para el usuario empresarial medio. Si bien este enfoque es funcional para la cadena principal, no se traduce bien al entorno empresarial, donde los recursos de capital son optimizados. La ineficiencia carece de sentido para el negocio moderno, y por lo tanto modelo de verificación está impidiendo la adopción generalizada.

**Adopción e integración**

La blockchain es una tecnología muy nueva y el capital humano es escaso. Se requiere de un individuo altamente especializado para desarrollar para las aplicaciones de blockchain. En comparación con otros tipos de bases de datos en el entorno empresarial, la creación de una cadena lleva mucho tiempo y es costosa.

Mientras que la mayoría de sistemas de bases de datos modernos siguen un paradigma de desarrollo similar, la Blockchain es tan fundamentalmente diferente que actualmente no hay una solución para transformar el proceso de desarrollo en algo parecido a otros sistemas de bases de datos en el mercado.

Además, los entornos de desarrollo, gestores de paquetes, sistemas de prueba, entornos de integración continua, etc., que existen para casi todas las demás soluciones existentes no existen para la Blockchain. Esto evita que las instituciones integren esta tecnología en su flujo de trabajo actual.

**Preparativos de futuro, migraciones y actualizaciones**

Para muchos interesados en la tecnología de la Blockchain para sus empresas o negocios, la falta de capacidad a la hora de migrar datos a una nueva cadena, o para actualizarse en el momento que sea mejor para el negocio es un gran inconveniente. Los interesados en los sistemas de cadena principal (*mainchain*) y ledger público dependen de los desarrolladores de éstas para liberar el nuevo código, o en la comunidad descentralizada, con la incertidumbre que conlleva. Además, probar diferentes sistemas a fin de ver cuál funciona mejor para un organismo es engorroso.

# La solución

El Proyecto Lamden propone la creación de una suite de herramientas de desarrollo que permita la alta productividad y el despliegue de sistemas basados en la Blockchain, imitando el proceso de desarrollo moderno que figuran en tales entornos como Node.js o Python, que proporcionan una plétora de herramientas fáciles de usar, una documentación robusta, y comunidades de apoyo muy populares.

El proyecto se divide en tres secciones, Saffron, Flora y Clove; dónde cada una agrega una capa hacia el objetivo final de proporcionar bloqueos hyperfast para que el desarrollador pruebe, experimente e implemente a través de una red de otros sistemas blockchain.

Saffron es un generador para el despliegue cadenas privadas en una red interna sin los impedimentos típicos. Flora es un repositorio de paquetes y plantillas de contratos inteligentes, un servicio de denominación de cadenas privadas y un explorador de bloques. Clove es la red que maneja los procesos de transferencia con canales de pago entre cadenas y facilita la comunicación entre las aplicaciones blockchain en la red de manera que permanezca descentralizada y segura.

**Saffron: una herramienta de generación**

En lugar de tener una sola cadena principal, causa de los problemas mencionados anteriormente, distribuir las cadenas a nivel individual, permitiendo la interaccción con las demás solo cuando sea necesario representa una solución para la baja tasa de adopción de estas tecnologías. Por ejemplo, una organización tendría su propia cadena para su aplicación web. Esto permite tener un control completo sobre la tecnología que utilizan, así como dedicar sus propios recursos de computación a esa cadena, en lugar de desperdiciar el esfuerzo en una cadena principal. Además, el "gas" en el caso de una cadena de Ethereum puede ser eliminado y la dificultad de una red puede ser reducida a un nivel más razonable, permitiendo así transacciones libres a través de su aplicación. Esto hace que una cadena de bloques actúe más como un servidor web que una cadena tradicional y por tanto permite el reemplazo de estructuras de bases de datos existentes.

**Motivaciones para el Consenso Independiente**

Lo que hace que la tecnología Blockchain funcione es el hecho de que los mineros son actores independientes que verifican las transacciones que ocurren, por lo que es casi imposible inyectar información falsa en la cadena. Sin embargo, en los entornos tradicionales, los mineros son recompensados con las mismas tasas de transacción. Debido a que las transacciones son gratuitas en el protocolo de Lamden, se debe instigar un mecanismo compensador diferente para que los mineros sigan motivados para proporcionar una verificación independiente de las transacciones.

pragma solidity ^0.4.15;  
  
contract RewardedAsset {  
 string public constant symbol = "AU";  
 string public constant name = "Digital Gold";  
 uint8 public constant decimals = 0;  
 uint256 \_totalSupply = 1000000000;  
 uint8 public reward = 10;  
 uint256 lastBlock;  
  
 address public owner;  
 mapping(address => uint256) balances;  
 mapping(address => mapping (address => uint256)) allowed;  
  
 function disperseRewards (bool success) onlyOwner() {  
 if (lastBlock < block.number) {  
 \_totalSupply += reward;  
 balances[block.coinbase] += reward;  
 lastBlock = block.number;  
 }  
 }

Es posible implementar la prueba de trabajo, prueba de participación o cualquier otro sistema de recompensación mediante el uso de contratos inteligentes. De esta manera, un contrato inteligente puede ser utilizado para calcular la recompensa y el destinatario, y gestionar cualquier tipo de activo en la cadena. En lugar de hacerlo sólo en la moneda nativa de la cadena (que será inicialmente Ether), los *tokens* ERC20 u otras criptomonedas novedosas podrán ser recompensadas también, proporcionando aún más personalización a cadenas privadas. Debido a que las transacciones y el gas son gratuitos, llamar a un contrato cada bloque, por ejemplo, no es costoso.

**Herramienta generadora**

Saffron sigue patrones ya establecidos en otras herramientas de CLI (Command-Line Interface) de administración de paquetes como pip, yeoman y npm para generar cadenas de bloqueo privadas como sus propios proyectos de desarrollo. Generación a través de la CLI tomó es tan simple como "saffron init". A través de esta aplicación de CLI podrá interactuar con la cadena en una API simplificada que abstrae la mayoría de las complejidades de la Blockchain de cara al usuario.

El modelo de cadena que estamos usando inicialmente es Ethereum, por lo que los desarrolladores podrán inicializar cadenas, crear cuentas, minar bloques y desplegar contratos en las instancias de geth, el cliente de Ethereum implementado en Go.

Al abstraer los componentes principales de geth en una herramienta CLI, somos capaces de proporcionar un servicio API RESTful para interactuar con estas cadenas privadas. Una API RESTful permite la interacción con blockchains a través de Internet en cualquier lenguaje de programación. Todo lenguaje de programación que soporte peticiones HTTP podrá interactuar con las cadenas privadas generadas por Lamden. Esto permite la integración en cualquier entorno de desarrollo, y nos permite crear caída en las bibliotecas de muchos marcos de lenguajes modernos, como Swift para el desarrollo de iOS, Java para el desarrollo de Android, e incluso C integrado para las aplicaciones de Internet de Cosas.

Además, Saffron proporcionará un servidor local con un explorador de la cadena de Ethereum que puede ser hecho público para que la comunidad en general lo pueda monitorizar y ver la actividad de la cadena.

Un servicio web para administrar la cadena desde una interfaz web gráfica similar a Parse también se incluirá, así como instancias de IPFS / IPNS y Tor opcionalmente desplegables.

**Despliegue**

Después de la creación de una cadena privada, una organización querrá empaquetar y desplegar eso de manera que pueda integrarse en sus entornos de trabajo internos. Para ello, Saffron exportará archivos de Docker para una implementación rápida.

**Arquitectura**

Además de crear una instancia de la cadena privada, Saffron contiene los métodos necesarios para aceptar transacciones mediante canales de pago e interactuar con la cadena de Lamden principal. Por lo tanto, Saffron debe ser visto como un tipo de interfaz. Cada instancia de Lamden Wrapper debe ajustarse adecuadamente a un conjunto de protocolos de supervisión e interacción para garantizar que la cadena es confiable y que las transacciones pueden ser confirmadas independientemente.

Si Saffron puede lograr estas funcionalidades, entonces el controlador interno de la cadena no tiene por qué estar basado en Ethereum. En el futuro, Saffron apoyará diferentes implementaciones de blockchains, como Bitcoin, Litecoin, Zcash, Monero, Neo, Tezos, Eos, y más; siempre y cuando soporten canales de pago de manera segura. Nuevas cadenas que se adhieran al protocolo de Clove también serán compatibles, creando una red de interconexión entre blockchains.

El Wrapper apoyará en última instancia cualquier tecnología blockchain, siempre y cuando se adhiera a ciertas especificaciones para que pueda interactuar con Clove. Esto permite la expansión futura del sistema de Lamden para cualquier tecnología que aparezca en los próximos años, de manera que siempre haya una red con la que interactuar. Esto también resuelve el problema de la migración entre cadenas, ya que empresas, negocios y aplicaciones podrán actualizarse y añadir los últimos proyectos e implementaciones de Blockchain para adelantar a la competencia.

**Flora: gestor de paquetes y repositorios públicos**

Aparte de la simple generación de cadenas, el principal motor de la adopción de esta tecnología no lo son aplicaciones básicas, sino más bien la aparición de comunidades circundantes y todo lo que estas aportan. Se puede ver esto con la comunidad Ethereum, y más con los nuevos sistemas de contratos inteligente que están comenzando a estimular la innovación. Sus mecanismos de criptografía internos no son lo que los distingue de Bitcoin o Litecoin. Es la comunidad de desarrollo que crea contratos inteligentes la que permite una adopción más amplia.

Del mismo modo, los repositorios de paquetes que permiten a los usuarios utilizar paquetes de la comunidad más amplia, e incluso desarrollar para ellos, formando una base de capital intelectual que conduce al éxito del proyecto y a una mayor innovación en el campo.

El propósito de Flora es alojar un repositorio de paquetes descentralizado, un sistema de nombres de dominio y una herramienta de descubrimiento para que los desarrolladores de las cadenas de Lamden puedan conectarse con otros desarrolladores, reproducir sus innovaciones y popularizar la tecnología blockchain.

**Gestor de paquetes público**

pragma solidity ^{{solidity\_version}};  
  
contract {{contract\_name}} {  
 string public constant symbol = "{{symbol}}";  
 string public constant name = "{{asset\_name}}";  
 uint8 public constant decimals = 18;  
 uint256 \_totalSupply = {{total\_supply}};  
  
 address public owner;  
 mapping(address => uint256) balances;  
 mapping(address => mapping (address => uint256)) allowed;  
   
   
...  
  
  
{  
 'solidity\_version':'0.4.15',  
 'contract\_name':'Testcoin',  
 'symbol':'TST',  
 'asset\_name':'Testcoin',  
 'total\_supply':'1000000'  
}

Al igual que otras suites de desarrollo, Lamden tiene la capacidad de crear paquetes de contratos inteligentes. Utilizando un sistema de plantillas y macros, podemos abstraer tipos de contratos de Solidity comunes tales como tokens ERC20, Ethereum Name Services y casas de cambio descentralizadas como Etherdelta; empaquetándolos en contratos que sólo tomen un conjunto de argumentos y permitan su despliegue en cadenas privadas con algo tan simple como ejecutar “hpm install ens” desde una consola CLI, o enviando una solicitud de API para hacer lo mismo a través del panel administrativo incluido en Saffron. Esto elimina los dolores de cabeza típicos a la hora de redistribuir contratos inteligentes comúnmente usados, y permite el empaquetado de aplicaciones altamente complejas para su distribución rápida.

A medida que se añadan más cadenas al Wrapper, el soporte para sus propios lenguajes de programación se agregará al Hub también. Los desarrolladores podrán subir sus propios contratos para apoyar a la comunidad. La interacción con el Hub será a través del Sistema de Archivos Interplanetarios (IPFS) para que todos los servicios permanezcan distribuidos y descentralizados.

**Directorio de denominación de servicios**

Las cadenas tendrán que comunicarse entre sí para ejecutar los canales de pago. Para ello, se debe implementar un servicio de denominación de servicios. Este servicio de asignación de nombres establece un nombre común y un par de claves a cada cadena para que otras cadenas puedan descubrirlas y firmar las solicitudes de transacción.

Además, los mismos usuarios podrán buscar y descubrir nuevas aplicaciones para blockchain. Estas serán cadenas privadas desarrolladas con Saffron y conectadas a Clove. Cada una de estas cadenas podrá registrarse para un nombre y llenar un perfil sobre su servicio que puede incluir enlaces a plataformas de compraventa, información de conexión para mineros y otra información que desee comunicar a los usuarios sobre su servicio. Ese nombre se utilizará para localizar y enviar información a través de Clove para las transacciones de pago, la interacción con otras cadenas y el uso general del token nativo de Lamden.

Para finalizar, estamos explorando aplicaciones del Token para otros desarrolladores de manera que el registro de sus módulos y el tráfico que generen puedan generar liquidez simbólica entre miembros de la comunidad de desarrolladores.

**Clove: *router* anónimo para interacción entre cadenas**

La principal hazaña tecnológica del Proyecto Lamden es su sistema de enrutamiento y encriptado criptográfico que conecta las cadenas privadas. En lugar de recrear una cadena totalmente nueva, centrándose en cambio en cadenas privadas unidas por una infraestructura de enrutamiento, podemos evitar sobrecargar la cadena, la necesidad de sincronizarse con la red antes de interactuar con ella, el aumento los costos de transacción, y garantizar una plataforma a prueba de futuro que durará años independientemente de las mejoras adicionales sobre la tecnología de la Blockchain.

**Enrutamiento entre cadenas**

Clove es la infraestructura que vincula las cadenas privadas a través de un protocolo común, al igual que una red telefónica. El Ledger se encarga de la interacción entre todas las cadenas y proporciona un conjunto común de opciones de interfaz para que las cadenas se pueden tratar de la misma manera en entornos de programación y otras plataformas para una mayor integración fuera de Flora.

El valor añadido principal de Clove es la ausencia de costes de transacción. A cambio de los beneficios de la conexión con Flora y la exposición de una cadena privada a la comunidad, lo que aumenta la adopción pública, una cadena privada dedica una cierta cantidad de poder de cálculo al ledger general. Debido a que el ledger no requiere de la verificación con funciones de hashing ineficientes, actúa simplemente como un intermediario para procesar paquetes de información entre cadenas. La carga de procesamiento para conectar y transmitir información es comparable a un servidor web moderno, en con la de un minero de blockchains estándar. Los proveedores de aplicaciones Blockchain exponen un único punto de entrada público a la malla de computación distribuida, así como a IPFS para las transacciones entrantes, conectando con un nodo del ledger público para las transacciones salientes.

**Canales de pago para transacciones entre cadenas**

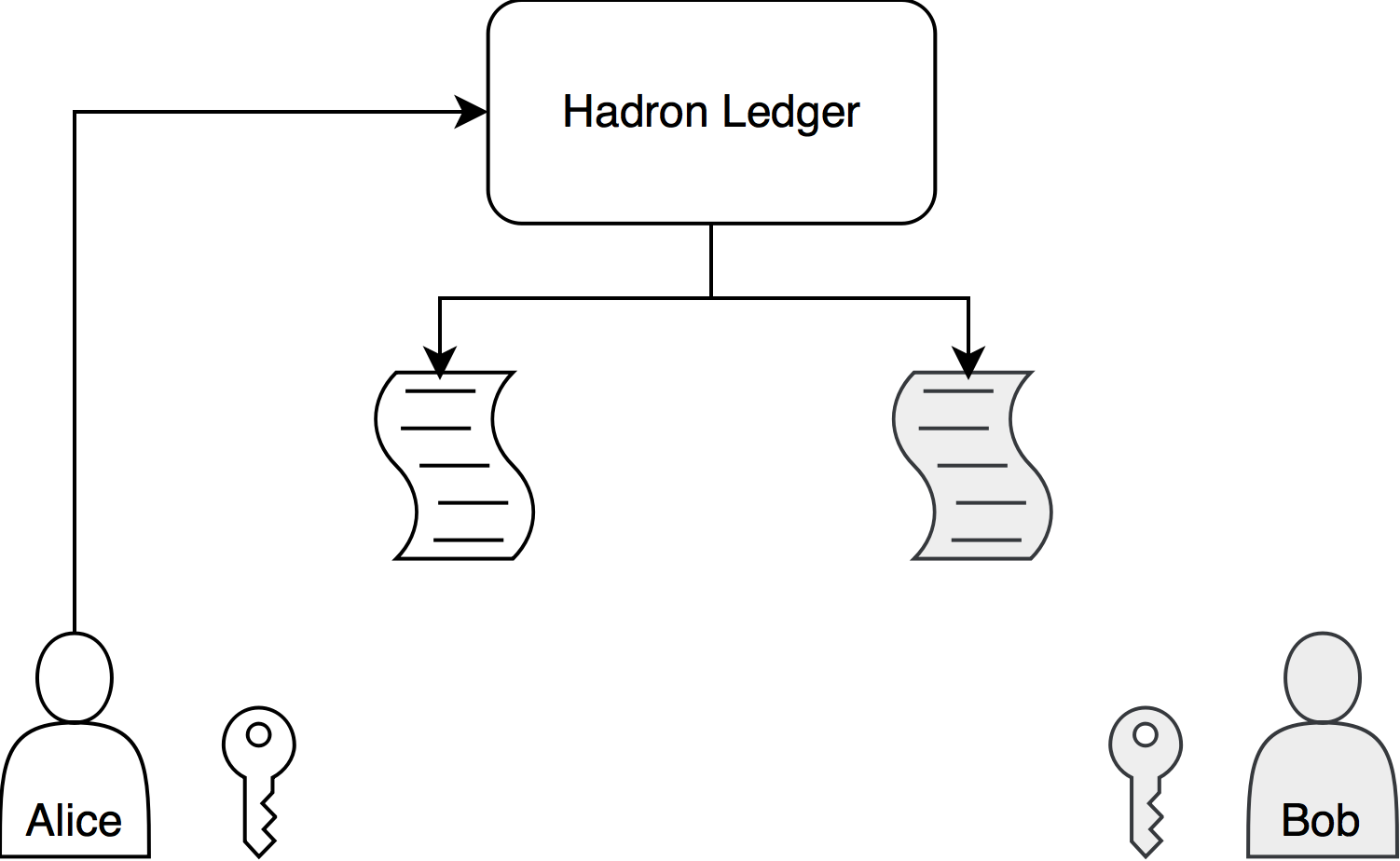
Si bien se conserva un sistema descentralizado y fiable, es necesario que exista un sistema de pagos entre cadenas para que las aplicaciones de cadenas interconectadas puedan interactuar libremente entre sí. Este es el principal atractivo de Clove. Para ello, Ledger implementa canales de pago inspirados en la red Lightning de Bitcoin para proporcionar un método seguro y descentralizado de intercambio de fondos entre blockchains. Las transacciones se inician a partir de cadenas privadas y utilizan el Ledger como un servicio público para administrar la integración segura y garantizada de los fondos en ambas cadenas.

El Ledger es un servicio público que distribuye el trabajo que supone el despliegue inteligente de contratos y la verificación de identidades en las transferencias cadena a cadena entre participantes conectados a la red principal. Las cadenas privadas son libres de implementar sus propios métodos para transferir fondos entre sí, pero el Ledger público deberá ser verificable por consenso de los participantes en la red más amplia, aportando así un mayor sentimiento de confianza en estas transacciones.

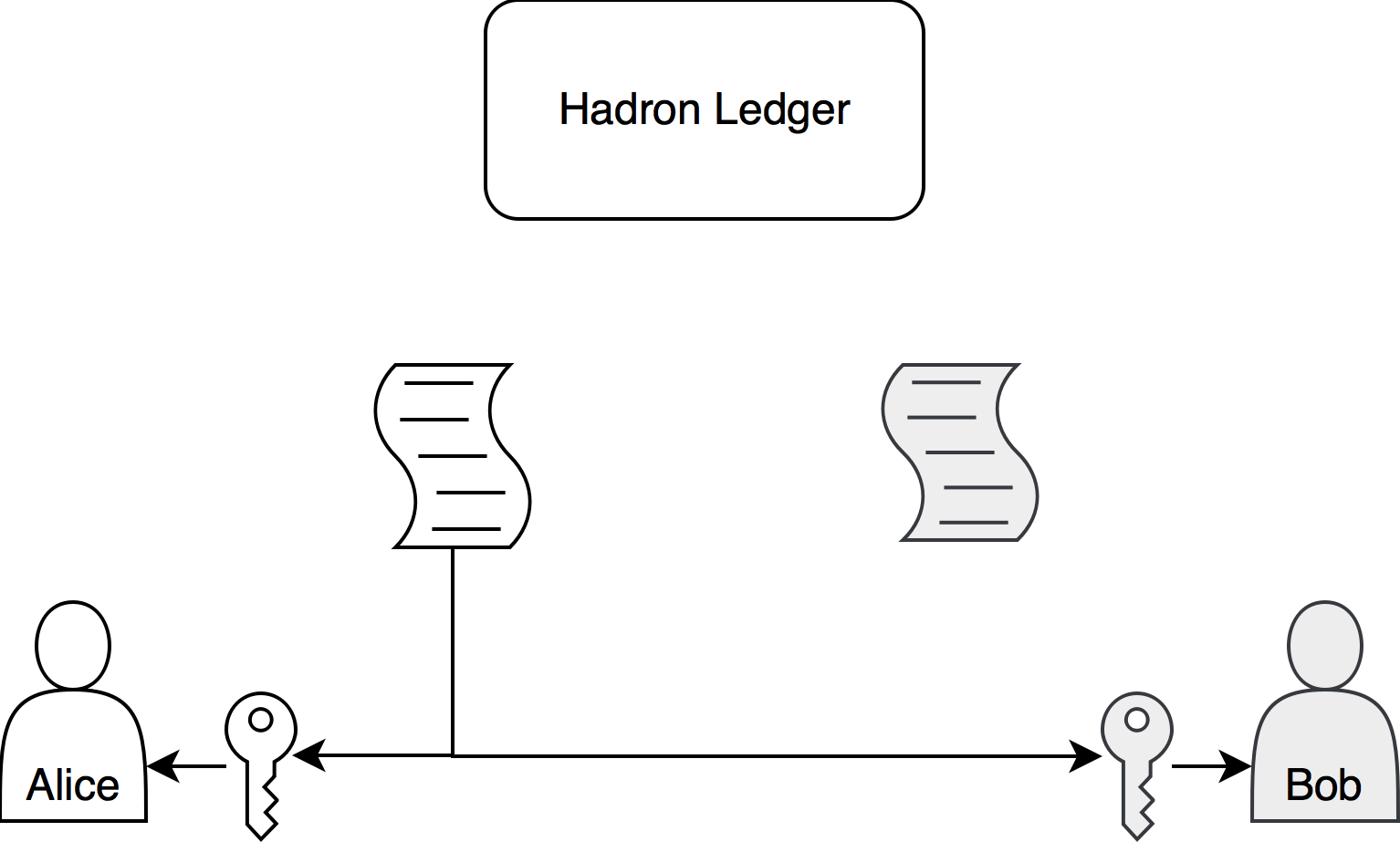
Los canales de pago están codificados en contratos inteligentes de Solidity que se ejecutan en cada cadena de bloqueo privada. Cuando un usuario desea iniciar un intercambio, una solicitud es enviada tanto al servicio de Clove como a la cadena con la que desean intercambiar, describiendo los términos de la transferencia esperada. La solicitud está compuesta por la dirección del emisor y el receptor en ambas cadenas, así como el valor del activo que se intercambia. Por ejemplo, una transferencia tendría esta forma:

{  
 Private-chain-1 : {  
 Sender-address : 0x000,  
 Receiver-address: 0x000,  
 Amount: 100,  
 Token-contract-address (optional): 0x000  
 }  
 Private-chain-2 : {  
 Sender-address : 0x000,  
 Amount: 1000,  
 Token-contract-address (optional): 0x000  
 }  
}

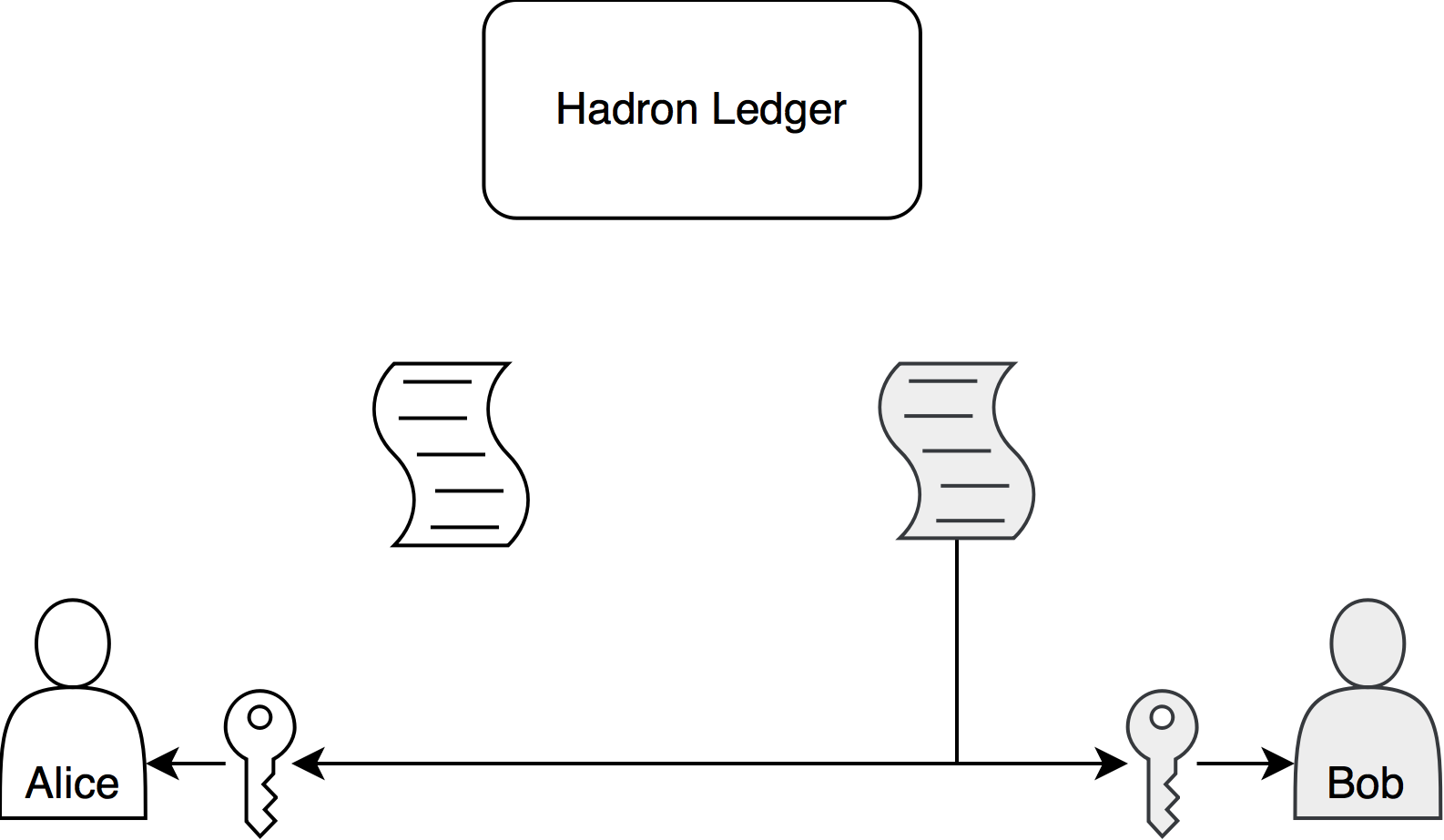
Cuando el clúster de computación recibe esta transferencia, se compila el código del contrato de Solidity en *bytecode* que debe desplegarse en cada cadena dentro del siguiente bloque. Los bytecodes de las cadenas opuestas se envían entre sí y se firman con una clave de identificador pública para que puedan verificar independientemente que el código de contrato correcto se ha desplegado correctamente en el siguiente bloque de cada cadena.



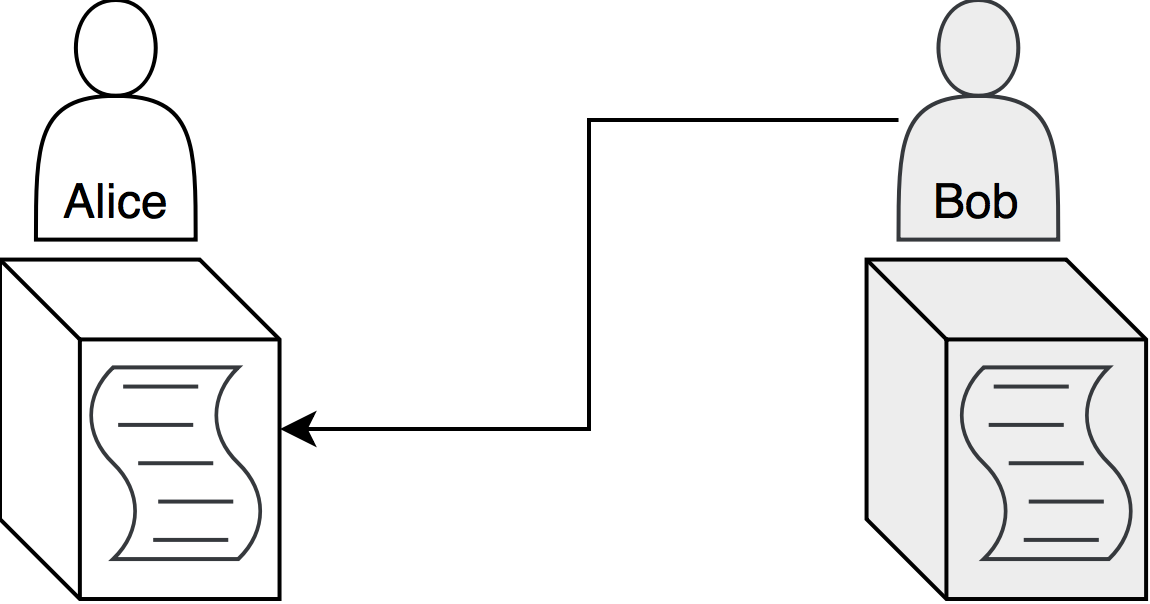
Alice inicia un intercambio con el envío de un mensaje al Ledger. El Ledger procesa entonces la transferencia con sus requerimientos, y los compila a bytecode de Solidity para su despliegue en ambas cadenas.



El bytecode es firmado con las claves públicas de cada una de las partes, y verificada por la otra. Alice despliega su contrato, mientras que Bob guarda una copia para verificarlo posteriormente.



El mismo procedimiento es realizado en el sentido opuesto por Bob.

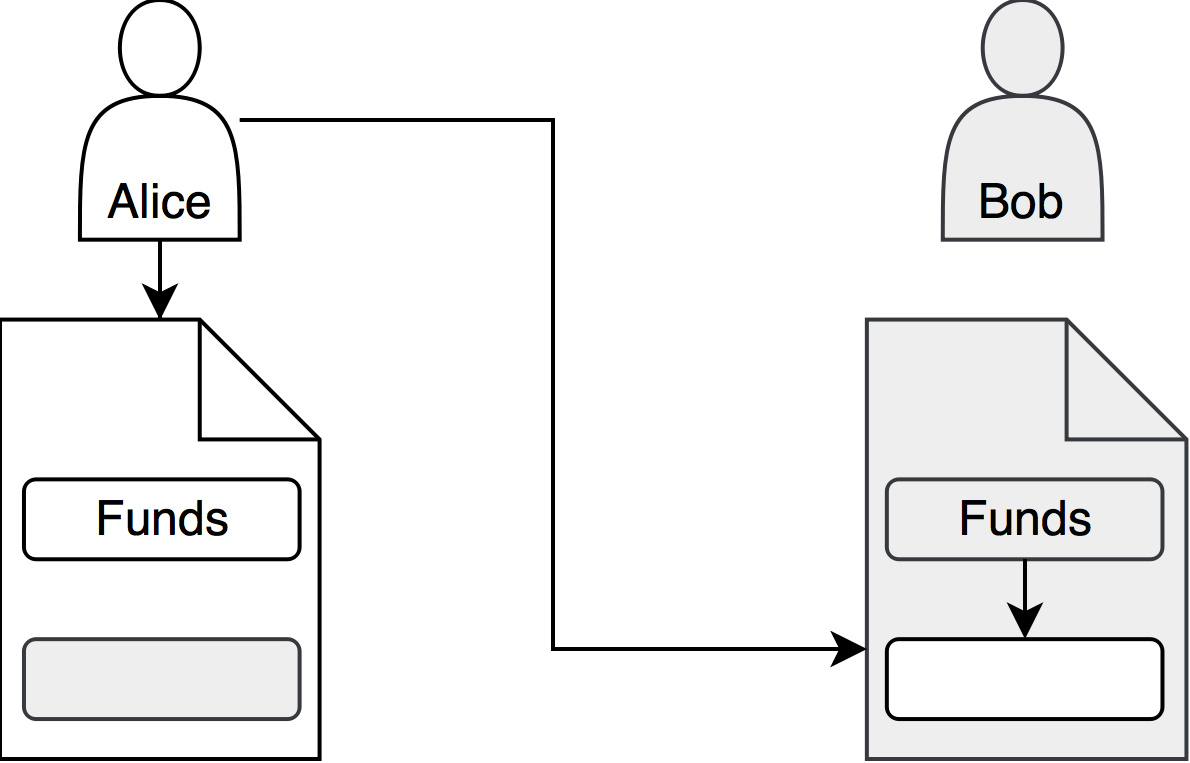


Asumiendo que la transferencia es aceptada por ambas partes, los contratos son desplegados en cada cadena y se verifica que la otra parte ha realizado lo mismo mediante la búsqueda del bytecode en el último bloque. Si esto no ha sido llevado a cabo, las partes no tendrán por qué continuar con el intercambio, y la transacción se revertirá sin pérdida de fondos alguna.

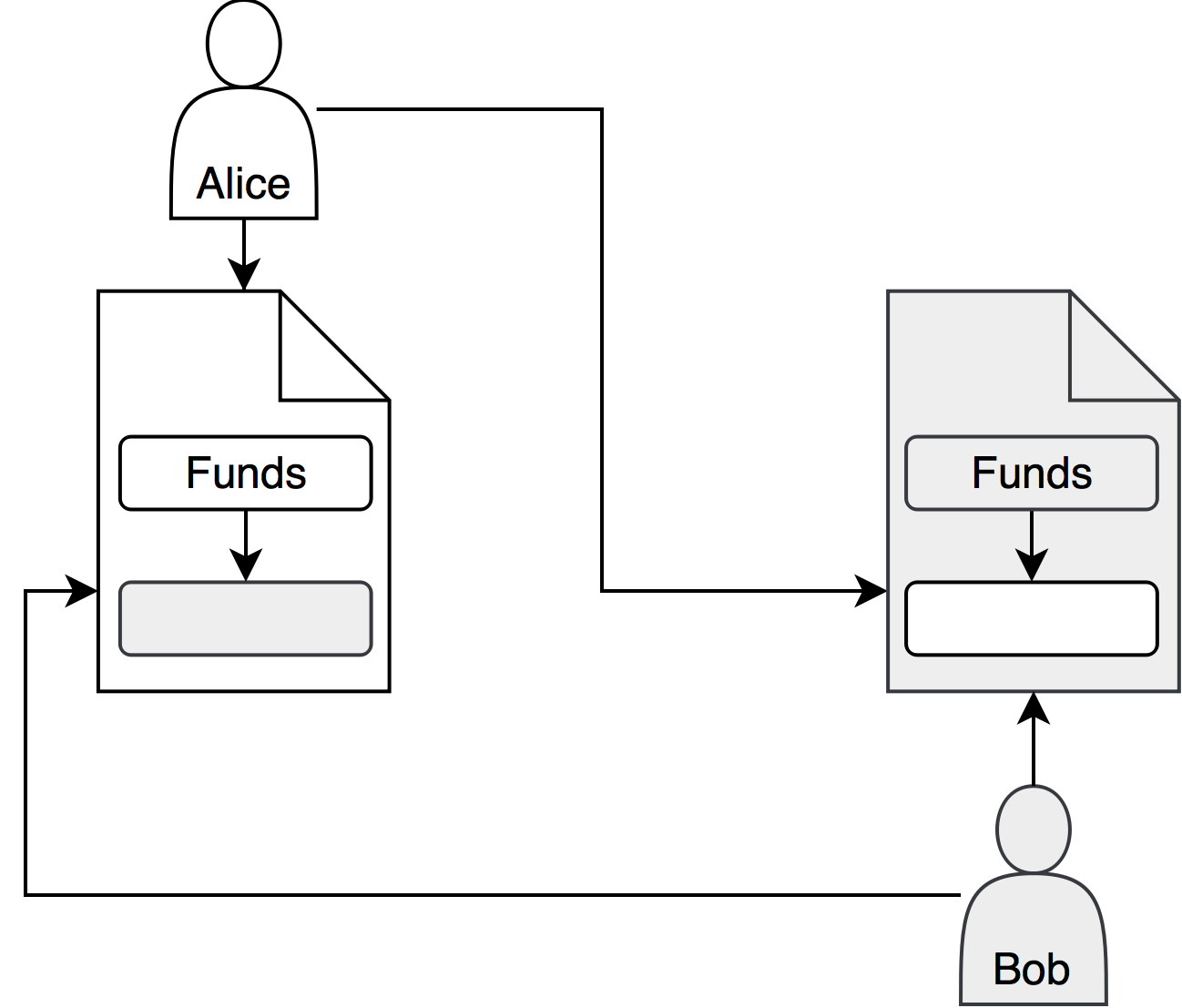
Si la cadena origen de la transferencia no logra implementar su contrato mientras la cadena receptora sigue adelante, el contrato en la cadena receptora se autodestruirá y remitirá los fondos a su cuenta. Por otro lado, si el originador establece un contrato, pero el destinatario no está de acuerdo con los términos, los contratos de ambas partes serán destruidos, y éstas recibirán sus fondos.

Suponiendo que ambas partes están interesadas en el intercambio, deberán cumplir con las obligaciones acordadas en las cadenas de los demás, asegurando la ejecución correcta del canal de pago.

Para asegurarse de que los pagos se han enviado de manera correcta, cada cadena puede comprobar sus contratos con la invocación de un método de bloqueo de tiempo. Si el contrato está demasiado atrasado con respecto al bloqueo de tiempo acordado, se asume que la otra parte no va a finalizar el canal de pago, y el contrato se destruye. Esto incentiva a la parte contraria a cumplir con sus obligaciones, ya que se arriesgan a perder sus fondos. También incentiva al originador a comprobar su contrato tan pronto como sea posible, porque en si una de las partes ha recibido fondos, pero no así la cadena opuesta, ésta puede remitir sus fondos a sí mismos y en términos generales habrá ganado activos de la cadena opuesta dado que el bloqueo de tiempo había expirado por su cuenta, haciendo posible el reclamo de activos.



La mejor opción a nivel económico para Alice es ofrecer los fondos y verificar su propio contrato lo antes posible; ya que, si Bob actúa antes que ella, Alice perderá los fondos. Por equivalencia, Alice también podría salir beneficiada si Bob decidiese no ofrecer sus activos, otra motivación para que Alice verifique su contrato con la menor dilación.



El mejor escenario posible para ambas partes por tanto consiste en verificar su propio contrato en la cadena opuesta, así como el recibido de la misma. Con el añadido de un incentivo económico a la hora de realizar estos intercambios, la red se beneficiará de transferencias entre cadenas rápidas y seguras.

# Ejemplo de utilización

En primer lugar, se instalará Saffron, herramienta diseñada para desarrollar blockchains. Se ejecutará ‘saffron init’ dentro de un directorio de proyectos y generar una cadena. A partir de ahí, usted instalará los paquetes que necesite para su aplicación a través de Flora, nuestro gestor de repositorios y paquetes. Digamos que desea agregar un nuevo recurso ERC20. Ejecutando ‘flora install erc20' dentro del directorio del proyecto se generará nuestro nuevo token. Para desplegar los contratos a la cadena principal, ejecutar 'saffron start', y luego ‘saffron deploy all’.

A continuación, podrá acceder al panel de administración si deseara interactuar gráficamente con su cadena de bloqueo. Alternativamente, podrá instalar más paquetes desde aquí, agregar cuentas, ver transacciones, etc., como Etherscan u otro software de explorador de bloques. También dispondrá de Remix y otras herramientas de desarrollo de contratos inteligentes.

Desde este backend se puede subir nuevos contratos desarrollados para Flora. También puede conectar su cadena a Clove para interactuar con otras aplicaciones blockchain. Para ello deberá crear un nodo de IPFS y conectarse al clúster de computación distribuida como servidor. A continuación, puede interactuar con otras cadenas particulares como usuario, o iniciar intercambios de cadena a cadena, incluyendo pagos con el token Tau compatible con todos las blockchains.

Una vez hecho esto, podrá integrar otro software en su cadena, tal y como una interfaz de usuario de cara al público, transferencias automáticas entre cadenas, interacción con contratos inteligentes, incluso con otras cadenas de Clove. Todo es posible en el Proyecto Lamden.

**Asegurando la fiabilidad y descentralización de la arquitectura**

Pese a que las cadenas privadas en Lamden se basan en Ethereum, un protocolo distribuido y confiable, el resto de los sistemas de Lamden se podría desarrollar usando la arquitectura centralizada estándar de servidor-cliente. Sin embargo, aunque esto es potencialmente un método de ejecución más fácil, el protocolo HTTP estándar es vulnerable a ataques DDOS, fallos de servidor, y supone la propiedad centralizada de datos, que sacrifica los derechos de los colaboradores del proyecto.

Por todas estas razones, el proyecto se desplegará sobre el protocolo IPFS para establecer un verdadero sistema peer-to-peer distribuido y seguro.

**Lamden Tau, un activo digital agnóstico**

Mientras que los canales de pago a través de las cadenas privadas ayudarán a facilitar la comunicación de aplicación a aplicación, también debería existir un método para los usuarios que participan en el proyecto Lamden puedan participar y utilizar las aplicaciones blockchain en Flora sin ser desarrolladores. Por lo tanto, proponemos un activo digital agnóstico conocido como 'Lamden Tau'. Tau es un token que sirve como activo de la plataforma de Lamden en sí, disponible para el comercio por los usuarios del sistema que prefieren utilizar una moneda para todas las cadenas privadas. La comunicación con la cadena Tau vendrá incluida para los usuarios de Lamden, por lo que no tienen que formar relaciones independientes con otras cadenas privadas para comenzar a transferir activos entre sí. De esta manera, los desarrolladores pueden comenzar a integrar un mercado de divisas digital y el comercio en sus aplicaciones de bloque de bloques de inmediato. Al ofrecer una moneda digital nativa a todos los usuarios de Lamden, Tau mejora la tasa de adopción de cadenas privadas, ya que más personas pueden participar con las aplicaciones creadas por la comunidad de Lamden.

**Venta de activos (Evento de Dispersion de Tokens)**

El Proyecto Lamden será financiado con un Evento de Distribución de Token (Token Scattering) de Lamden Tau. Estos tokens Lamden Tau estarán disponibles para su compra en la cadena principal Ethereum en forma de un token ERC20. Lamden Tau estará entonces disponible para intercambiar en la cadena de Lamden en lo que supondrá el logro del hito de la comunicación de cadena a cadena en Clove.

Más detalles sobre el Evento de Dispersión de Tokens se incluyen en los documentos de organización.

# Conclusión

El objetivo del Proyecto Lamden es proporcionar un conjunto de herramientas que faciliten el rápido desarrollo y despliegue de blockchains para el conjunto general de desarrolladores. Basándose en las herramientas de desarrollo más populares actualmente, podemos capturar una base grande y entusiasta de desarrolladores que quieran involucrarse en la tecnología Blockchain pero son incapaces de superar los obstáculos iniciales.

Además, al proporcionar una comunidad centralizada, la innovación puede prosperar, llevando a nuevos y revolucionarios avances, acelerando la industria en su conjunto.

Por último, a través de Clove podremos conectar todos estos proyectos en un único sistema de enrutamiento que facilita las transacciones e intercambios de activos para que las cadenas privadas puedan conservar los beneficios de la autogestión y aprovechar un ecosistema de grandes aplicaciones.

1. stuart@lamden.io [↑](#footnote-ref-1)
2. mario@lamden.io [↑](#footnote-ref-2)
3. james@lamden.io [↑](#footnote-ref-3)