

ICON 2.0: Introducing a new blockchain software architecture based on Go

Batang: The Foundation for the Future Growth of the ICON Network

Attention ICONists,

Our biggest secret is finally out. Today, we are excited to announce our next significant technical advancement — **ICON 2.0: BATANG, a new blockchain software architecture based on Go**. The word “Batang” means “base” or “foundation” in the Korean language and we see ICON 2.0 as the base or foundational technology that will support the future growth of the ICON network.

The current ICON mainnet, built entirely on Python, was our first raw attempt to create and experiment with a public blockchain network. We were proud that ICON became one of the top blockchain networks, but we knew we could do much better now with more experience under our belt. Therefore, about a year ago, ICON’s core developers started to completely redesign the existing loopchain’s architecture in an effort to prepare us for the future. Now, the ICON 2.0 is built and ready to go.

ICON 2.0 is a brand new software; a faster, more stable, and scalable high-performance blockchain that inherits the current ICON Network. We are completely overhauling the ICON mainnet that was launched in 2018. **ICON 2.0 will come standard with interoperability features to support and power cross-chain DeFi solutions.** At launch, the ICON Foundation will deploy all necessary smart contracts on high-profile blockchains and will also run the relayers.

ICON 2.0 uses an enhanced and completely rewritten blockchain engine: “Goloop”, provi-

ICON 2.0: Presentamos una nueva arquitectura de software basada en Go

Batang: La Base para el Futuro Crecimiento de la Red ICON

Atención, ICONistas,

Nuestro mayor secreto por fin ha sido revelado. Hoy, estamos emocionados de anunciar nuestro siguiente avance tecnológico — **ICON 2.0: BATANG, una nueva arquitectura de software basada en Go**. El término «Batang» significa «base» o «pilar» en idioma coreano y vemos ICON 2.0 como la tecnología básica o fundamental que sostendrá el crecimiento futuro de la red ICON.

La *mainnet* actual de ICON, construida en su totalidad sobre Python, fue nuestro primer intento de crear y experimentar con una red pública basada en *blockchain*. Estuvimos orgullosos de que ICON se convirtiera en una de las principales redes *blockchain*, pero sabíamos que lo podríamos hacer mucho mejor ahora con más experiencia en nuestro bagaje. Por consiguiente, hace alrededor de un año, los desarrolladores del *core* de ICON comenzaron a rediseñar completamente la arquitectura de *loopchain* existente en un esfuerzo para prepararnos para el futuro. Ahora, la red ICON 2.0 está finalizada y lista.

ICON 2.0 es un software totalmente nuevo; una *blockchain* de alto rendimiento más rápida, estable y escalable que sustituye a la actual Red ICON. Estamos reestructurando completamente la *mainnet* de ICON que fue lanzada en 2018. **ICON 2.0 será el estándar en funcionalidades de interoperabilidad para promover y potenciar soluciones DeFi cross-chain.** Al lanzamiento, la Fundación ICON desplegará todos los *smart contracts* necesarios en *blockchains* de alto rendimiento y también pondrá en marcha los *relayers*.

ICON 2.0 usa un motor de *blockchain* completamente reescrito y mejorado: «Goloop»,

ding an improved blockchain experience over the existing Python-based loopchain.

Again, we would like to emphasize that goloop base modules for enterprises are completed. Some of ICONLOOP's enterprise clients including Seoul Metropolitan Government are also preparing to use the new blockchain engine. We are planning a smooth migration process and there should be no necessary actions to be taken by ICONists. We will share more details as the implementation plans roll out.

New Features in ICON 2.0

One of the most exciting aspects of ICON 2.0 is it contains a large number of enhanced core features needed for our community, and an opportunity to redesign some of the current blockchain's features. Here we will share all of the features and opportunities for ICON 2.0. Some of these features will be discussed on the forum amongst P-Reps to come to consensus on the finer details, and many of these features still require more research.

Python SCORE Executor

Python SCORE Executor provides a pure Python execution environment that will be operated in a separate process from the consensus engine. It can execute the already deployed Python SCOREs on the ICON network as it is. By separating the executor process from the consensus engine, we can handle infinite loop and instability issues of Python SCOREs.

Java SCORE Executor

Now SCORE developers can write their program using the Java programming language. SCOREs written in Java would run on the Java virtual machine, thus SCOREs can be executed securely and stably without requiring the audit process, which has been a major

ofreciendo una experiencia de *blockchain* mejorada frente a la *loopchain* existente basada en Python.

De nuevo, nos gustaría subrayar que los módulos fundamentales de goloop para empresas ya están completados. Algunos de los clientes corporativos de ICONLOOP, incluyendo al Gobierno Metropolitano de Seúl, también se están preparando para usar el nuevo motor de *blockchain*. Preve-
mos un proceso fluido de migración y no debería ser necesaria ninguna acción por parte de los ICONistas. Compartiremos más detalles a medida que los planes de implementación se vayan desarrollando.

Nuevas Funcionalidades en ICON 2.0

Uno de los aspectos más emocionantes de ICON 2.0 es que contiene un gran número de funcionalidades del *core* mejoradas, necesarias para nuestra comunidad, así como que supone una oportunidad para rediseñar varias de las funcionalidades de *blockchain* actuales. A continuación enumeramos todas las funcionalidades y oportunidades que se debatirán en el foro entre P-Reps para llegar a un consenso sobre los detalles, muchas de las cuales todavía requerirán de una mayor investigación.

SCORE Executor en Python

El SCORE Executor en Python proporciona un entorno puro de ejecución Python que funcionará en un proceso separado al del motor de consenso. Puede ejecutar los SCOREs en Python ya desarrollados en la red ICON tal cual. Separando el proceso de ejecución del motor de consenso, podemos manejar infinitos *loops* y problemas de estabilidad de los SCOREs Python.

SCORE Executor en Java

Ahora, los desarrolladores de SCORE pueden desarrollar su programa en lenguaje de programación Java. Los SCOREs escritos en Java funcionarían en la máquina virtual Java, de modo que tales SCOREs pueden ser ejecutados de manera segura y estable sin

pain point for developers on the current ICON mainnet. Since Java SCOREs don't require an audit, we'll be encouraging future developers to use the Java SCOREs. Furthermore, Java SCOREs can be interoperable with the existing Python SCOREs through inter-SCORE calls, which makes for a smooth transition to the Java SCORE environment.

New P2P protocol

A new protocol to synchronize the state between nodes will be integrated. A new node uses both gossip and multicast protocol to deliver messages. This requires a structured network supported by community members. In most cases, messages are delivered through multicast protocol using redundant paths, but they use gossip protocol in some exceptional cases like discovering the last state, recovering missed messages, and so on.

Fast Sync

Normally nodes need to synchronize all of the historical blockchain data before joining the consensus or querying the last state. But most users are not interested in historical data. For those users, we are planning to support the Fast Sync feature. If it's enabled, they can provide most services, except querying old transactions, in a fairly short time. DApps using historical data do not use this feature. Representative nodes may use this feature for a fast start-up, but they need to synchronize all the historical data.

Object Merkle Patricia Tree

Most merkle tree implementations calculate hashes of stored data at adding an entry. And also they provide just an interface to store bytes. Object Merkle Patricia Tree (OMPT) calculates hashes only when they're required; until then it manages all data as immutable objects. With this scheme, it calculates the

necesidad de un proceso de auditoría, lo que ha venido suponiendo el mayor quebradero de cabeza para los desarrolladores en la *mainnet* actual de ICON. Dado que los SCOREs en Java no requieren de auditoría, animamos a los futuros desarrolladores a usar los SCOREs Java. Asimismo, los SCOREs de Java pueden interactuar con los SCOREs de Python existentes a través de llamadas inter-SCORE, lo que supone una transición fluida al entorno de SCORE en Java.

Nuevo protocolo P2P

Se va a integrar un nuevo protocolo para sincronizar el estado entre nodos. Un nuevo nodo utiliza tanto protocolos *gossip* como *multicast* para enviar mensajes. Esto requiere de una red estructurada sostenida por miembros de la comunidad. En la mayoría de los casos, los mensajes se envían a través del protocolo *multicast* utilizando rutas redundantes, pero usan el protocolo *gossip* en algunos casos excepcionales como para descubrir el último estado, recuperar mensajes perdidos, etc.

Sincronización Rápida

Normalmente los nodos deben sincronizar todos los datos históricos de la *blockchain* antes de unirse al consenso o hacer un *query* del último estado; pero a la mayoría de usuarios no les interesan los datos históricos. Para estos usuarios, estamos planeando dar soporte a la funcionalidad de Sincronización Rápida. Si se activa, podrán prestar la mayoría de servicios -excepto hacer *query* sobre transacciones antiguas- en un corto espacio de tiempo. Las DApps que usen datos históricos no usarán esta funcionalidad. Los nodos de representación pueden usar esta funcionalidad para un arranque rápido, pero tienen que sincronizar todos los datos históricos.

Árbol de Merkle-Patricia de Objeto

La mayoría de las implementaciones del árbol de Merkle calculan los *hashes* de datos almacenados al añadir una entrada. También proporcionan simplemente una interfaz para almacenar bytes. El Árbol de Merkle-Patricia de Objeto (OMPT, por sus siglas en inglés) calcula los *hashes* sólo cuando es necesario;

hashes at the end of the execution of all transactions in the block.

Performance Enhancements

With Python implementation, it's hard to utilize multi-cores using multiple threads because of the global interpreter lock (GIL). Go provides goroutines to manage threads efficiently. Although the runtime supports garbage collection, it does not make any big response delay for collecting garbage. It reduces response time in handling most user requests and enables handling more user requests concurrently compared with Python implementations.

BTP

BTP is a general purpose interoperability protocol, however, it will come standard on ICON 2.0 with a specific initial use case in mind. We will be supporting interoperability with other public blockchains in order to support cross-chain DeFi solutions. At launch, the ICON Foundation will deploy all necessary smart contracts on high-profile blockchains and will also run the relayers, however, any individual or group may run a private relayer with their own fee system.

ICX holders will have the opportunity to pre-register a relayer in preparation for a decentralized relayer network. There will be a minimum ICX stake requirement to pre-register, and pre-registered relayers will earn any inflation allocated to relayers and the fees generated by cross-chain transactions based on the amount of ICX they have staked. The fees earned will be paid in the asset which was sent, meaning that if somebody sends ETH to the ICON blockchain, Relayers will earn ETH as fees.

Vote Spreading

hasta entonces gestiona los datos como objetos inmutables. Siguiendo esta fórmula, calcula los *hashes* al final de la ejecución de todas las transacciones en el bloque.

Mejoras de Rendimiento

Con la implementación de Python, es difícil utilizar *cores* múltiples usando múltiples *threads* debido al bloqueo de intérprete global (GIL). Go proporciona rutinas «go» para gestionar los *threads* de manera eficiente. Si bien el tiempo de ejecución soporta la recolección de desechos, no conlleva un gran retardo de respuesta para recolectar los mismos. Reduce los tiempos de respuesta en procesar las peticiones de la mayoría de los usuarios y posibilita la gestión de más peticiones concurrentes de usuarios comparado con las implementaciones en Python.

BTP

BTP es un protocolo de interoperabilidad de propósito general, si bien se convertirá en el estándar en ICON 2.0 teniendo en cuenta un caso de uso inicial específico. Mantendremos la interoperabilidad con otras *blockchains* públicas para dar soporte a soluciones DeFi entre diferentes cadenas. A su lanzamiento, la Fundación ICON desplegará todos los contratos inteligentes necesarios en *blockchains* de perfil alto y también ejecutará los *relayers*, si bien todo individuo o grupo podrá ejecutar un *relayer* privado con su propio sistema de comisiones.

Los poseedores de ICX tendrán la posibilidad de pre-registrar un *relayer* en previsión de una red de *relayers* descentralizados. Habrá un requisito de *stake* mínimo de ICX para el pre-registro, y los *relayers* pre-registrados ganarán cualquier inflación asignada a los mismos y las comisiones generadas por transacciones entre cadenas en función de la cantidad de ICX que tengan en *staking*. Las comisiones se pagarán en el activo en que fueran enviadas, queriendo esto decir que, si alguien envía ETH a la blockchain de ICON, los *Relayers* ganarán ETH como comisión.

Difusión de Votos

Vote Spreading is a novel solution for systematically decentralizing a DPoS network, where inactive voters have their ICX spread to all top 100 P-Reps. This will solve the issue of vote stagnancy and allow active ICX holders to have the greatest impact on governance. Further details on vote spreading will be shared in a separate post and discussed on the forum.

IISS 3.1

ICON 2.0 gives us more freedom to design a cleaner and more easily understood economic design. The basic structure of IISS 3.1 follows the structure of IISS 3.0 already discussed in the community. However, the IISS 3.1 design simply divides inflation into a few different pre-defined categories. For example:

P-Reps: 17.5%
Relayers: 2.5%
Contribution Proposal Fund: 10%
Voters: 70%

ICON 2.0 will include a network proposal to allow P-Reps to easily adjust these inflation allocations using an on-chain and self-executing vote. More details regarding IISS 3.1 will be shared in a separate post and discussed on the forum.

Multi-Channel Technology

Multi-channel is a form of scalability, where each DApp on ICON could instead be an application specific channel (see *Band Protocol* and *Kava* for examples of application specific chains using the Cosmos SDK). Each channel is essentially its own blockchain, sparing DApp developers platform risk while making it easy to launch their own network. The technology behind channel chains is complete for ICON 2.0, however, additional work will need to be done to productize the software after the successful migration.

With this strategy in mind, we would like to turn the ICON Nexus (the primary network operated

La Difusión de Votos es una solución novedosa para descentralizar de manera sistemática la red DPos, en la cual los votantes inactivos asignan su difusión de ICX a todos los principales 100 P-Reps. Esto solucionará el problema del estancamiento de votos y permitirá a los poseedores activos de ICX tener el mayor impacto en la gobernanza. Más detalles sobre la difusión de votos se desvelarán en una entrada separada y debatidos en el foro.

IISS 3.1

ICON 2.0 nos proporciona más libertad para diseñar un modelo económico más limpio y fácilmente comprensible. La estructura básica de IISS 3.1 sigue la estructura del ya debatido por la comunidad IISS 3.0. Sin embargo, el diseño de IISS 3.1 simplemente divide la inflación en unas pocas categorías predefinidas distintas. Por ejemplo:

P-Reps: 17.5%
Relayers: 2.5%
Fondo de Propuesta de Contribución: 10%
Votantes: 70%

ICON 2.0 incluirá una propuesta de red para permitir a los P-Reps ajustar fácilmente estas asignaciones de inflación usando un voto en la cadena y auto-ejecutable. Más detalles sobre IISS 3.1 se desvelarán en una entrada separada y debatidos en el foro.

Tecnología Multi-Canal

La Multi-Canalidad es una forma de escalabilidad, en la que cada DApp en ICON podría ser un canal específico de aplicación (ver *Band Protocol* y *Kava* para ejemplos de aplicación de cadenas específicas usando el SDK Cosmos). Cada canal es esencialmente su propia *blockchain*, moderando el riesgo de plataforma de desarrolladores de DApp mientras que facilita el lanzamiento de su propia red. La tecnología detrás de las cadenas de canal es completa para ICON 2.0, si bien hará falta trabajo adicional para convertir en producto el software tras la migración exitosa.

Con esta estrategia en mente, nos gustaría convertir el Nexus de ICON (la red principal

by P-Reps) into the security, immutability, and governance layer of the ICON Republic and encourage future DApps to launch application specific channels. In practice, this means increasing the main P-Rep set significantly to create more decentralization, more security, and more immutability. Channel chains would have fewer nodes in order to be more scalable, but would periodically backup their data on the ICON Nexus (secure, decentralized, immutable) using a BTP connection.

Fully open source development process

In order to make the development process completely transparent, and to share all the progress with the community, we decided to share all the development processes on Github at the beginning.

On this Github, you can see the source code of the next generation loopchain based on Go, called “Goloop”, which as mentioned, we’ve developed for over a year. Any community member can verify the code and technology of the ICON team on this Github Repository. We are confident that this development direction will be helpful to the community to understand the structure of ICON 2.0 in the long term.

Github Repository:

<https://github.com/icon-project/goloop>

Currently, there are two branches in the repository. “master” is the development branch for ICON 2.0 and the “base” is the source code of goloop. Therefore, the updates only focused on the goloop base modules will be committed to the “base” branch, while the economics and governance logics focused updates are committed to the “master” branch.

More technical docs and info will be uploaded on Github soon.

New development strategy focused

operada por P-Reps) en una capa de seguridad, inmutabilidad y gobernanza de la República ICON y promover que las DApps lancen canales específicos de aplicaciones. En la práctica, esto supone un aumento significativo del conjunto de P-Reps para crear una mayor descentralización, mayor seguridad y mayor inmutabilidad. Las cadenas de canales tendrían menos nodos para ser más escalables, pero realizarían copias de seguridad de datos periódicas en el Nexus de ICON (seguro, descentralizado, inmutable) usando una conexión BTP.

Proceso de desarrollo en código totalmente abierto

De cara a volver el proceso de desarrollo totalmente transparente, y compartir todo el progreso con la comunidad, hemos decidido compartir todos los procesos de desarrollo en Github desde el principio.

En este Github, puedes ver el código abierto de la próxima generación loopchain basada en Go, llamada «Goloop» que, como hemos mencionado, llevamos desarrollando por más de un año. Cualquier miembro de la comunidad puede verificar el código y la tecnología del equipo de ICON en este Repositorio de Github. Confiamos en que esta dirección de desarrollo será beneficioso para que la comunidad entienda la estructura de ICON 2.0 a largo plazo.

Repositorio Github:

<https://github.com/icon-project/goloop>

Actualmente, hay dos ramas en el repositorio; «master» es la rama de desarrollo para ICON 2.0 y «base» es el código abierto de goloop. Así, las actualizaciones enfocadas exclusivamente en los módulos básicos de goloop dependerán de la rama «base», mientras que la lógica económica y de gobernanza enfocada en actualizaciones dependerá de la rama «master».

Se subirá más documentación e información técnica a Github pronto.

Nueva estrategia de desarrollo enfo-

on ICON 2.0

Since we are migrating to ICON 2.0, we have made the decision to save some portions of development for ICON 2.0 rather than duplicating work, introducing potential instability in ICON 1.0, and slowing down the entire migration process. You may have noticed there was not much conversation regarding upgrades to the network after IISS 3.0, and that is because we were discussing whether or not to migrate to ICON 2.0 or spend time enhancing the current ICON Network.

We have decided to implement only the Contribution Proposal System on the current ICON Network, but all other planned network enhancements will be saved for ICON 2.0. Additionally, instead of being funded by block rewards (because this creates significant extra work), ICON will be using a portion of our block rewards to fund the CPS while still keeping a portion to continue funding grants.

This is an exciting time to be an ICONist, and we look forward to your participation and feedback as we roll out ICON 2.0 and continue building one of the most advanced, decentralized blockchain networks in the world.

Thank you,

The ICON team

ICON official community

Homepage : <https://icon.foundation>

Medium (ENG) : <https://medium.com/helloiconworld>

Brunch (KOR) : <https://brunch.co.kr/@helloiconworld>

KakaoTalk (KOR) : <https://open.kakao.com/o/gMAFhdS>

Telegram (ENG) : https://t.me/hello_iconworld

Telegram (KOR) : <https://t.me/iconkorea>

cada en ICON 2.0

Como estamos migrando a ICON 2.0, hemos tomado la decisión de aprovechar algunas iteraciones de desarrollo para ICON 2.0 en vez de duplicar el trabajo, causando inestabilidad potencial en ICON 1.0 y ralentizando el proceso completo de migración. Puede que hayas notado que no ha habido demasiada conversación acerca de las actualizaciones en la red tras el IISS 3.0, y es debido a que hemos estado debatiendo si migrar o no a ICON 2.0 o emplear el tiempo en mejorar la red ICON actual.

Hemos decidido implementar solamente el Sistema de Propuestas de Contribución (CPS) en la red ICON actual, pero todas las mejoras de red planeadas se guardarán para ICON 2.0. Además, en vez de financiarse mediante recompensas de bloque (y porque genera un significativo trabajo extra), ICON usará una parte de nuestras recompensas de bloques para financiar el CPS a la par que se conservará una parte para seguir financiando ayudas.

Son tiempos emocionantes para ser ICONista, y esperamos tu participación y *feedback* a medida que lancemos ICON 2.0 y continuemos construyendo una de las redes *blockchain* más avanzadas y descentralizadas del mundo.

Gracias,

El equipo de ICON

Comunidad oficial de ICON

Página Principal: <https://icon.foundation>

Medium (inglés): <https://medium.com/helloiconworld>

Brunch (coreano): <https://brunch.co.kr/@helloiconworld>

KakaoTalk (inglés): <https://open.kakao.com/o/gMAFhdS>

Telegram (inglés): https://t.me/hello_iconworld

Facebook : <https://www.facebook.com/helloicon/>

Reddit : <https://www.reddit.com/r/helloicon/>

Twitter (Foundation) : <https://twitter.com/helloiconworld>

Twitter (Republic) : <https://twitter.com/IconRepublic>

F.A.Q Portal : <https://helloiconworld.freshdesk.com/support/home>

Contact : hello@icon.foundation

Telegram (coreano): <https://t.me/iconkorea>

Facebook: <https://www.facebook.com/helloicon/>

Reddit: <https://www.reddit.com/r/helloicon/>

Twitter (Fundación): <https://twitter.com/helloiconworld>

Twitter (República): <https://twitter.com/IconRepublic>

Portal F.A.Q (Preguntas Frecuentes): <https://helloiconworld.freshdesk.com/support/home>

Contacto: hello@icon.foundation